

VNU-HUS MAT3500: Toán rời rạc

Tập hợp bài giảng

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn

Ngày 6 tháng 12 năm 2025

Nội dung

1. Logic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Đệ quy
4. Thuật toán I: Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp
5. Thuật toán II: Thuật toán tham lam, thuật toán đệ quy
6. Lý thuyết số cơ bản
7. Các phương pháp đếm
8. Lý thuyết đồ thị I: Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông
9. Lý thuyết đồ thị II: Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị
10. Lý thuyết đồ thị III: Cây
11. Đại số Boole

1. Logic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Độ quy
4. Thuật toán I: Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp
5. Thuật toán II: Thuật toán tham lam, thuật toán đệ quy
6. Lý thuyết số cơ bản
7. Các phương pháp đếm
8. Lý thuyết đồ thị I: Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông
9. Lý thuyết đồ thị II: Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị
10. Lý thuyết đồ thị III: Cây
11. Đại số Boole

VNU-HUS MAT3500: Toán rời rạc

Lôgic và Chứng minh

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn



Nội dung



Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Lôgic mệnh đề

Mệnh đề



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

2

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Mệnh đề

Một *mệnh đề (proposition)* là một phát biểu đúng (True) hoặc sai (False), chứ không thể vừa đúng vừa sai

- ✓ Hà Nội là thủ đô của Việt Nam
- ✓ $1 = 2$
- ✓ $1^3 + \dots + 9^3 = (20 + 25)^2 = 2025$
- ✓ Mọi số chẵn lớn hơn hoặc bằng 4 là tổng của hai số nguyên tố (Giả thuyết Goldbach)

- ✗ Máy giờ rồi?
- ✗ Hãy đọc quyển sách này
- ✗ Thời tiết hôm nay lạnh quá
- ✗ $x + 1 = 2$

[Câu hỏi]

[Mệnh lệnh]

[Ý kiến, Cảm thán]

[Đúng/sai tùy vào x]

Lôgic mệnh đề

Mệnh đề



Bài tập 1

Câu nào sau đây là một mệnh đề?

- (1) Trái Đất là một hành tinh
- (2) $1 + 2$
- (3) $1 + 2 = 3$
- (4) Hôm nay trời mưa
- (5) Bạn có nói tiếng Anh không?
- (6) $x + y = 5$
- (7) A ha ha ha ha
- (8) Hãy đưa cho tôi quyển sách kia
- (9) Rất tốt!
- (10) Nếu $x = 3, y = 4, z = 5$ thì $x^2 + y^2 = z^2$

Bài tập 2

Khẳng định “Phát biểu này là sai” có phải là một mệnh đề lôgic hay không? Vì sao?

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

3

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Lôgic mệnh đề

Toán tử lôgic và bảng chân trị



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

4

- Ta thường sử dụng các chữ cái $p, q, r, s, \dots$ để ký hiệu các mệnh đề
- Mệnh đề đúng có *giá trị chân lý đúng T* (True). Mệnh đề sai có *giá trị chân lý sai F* (False)
- *Mệnh đề phức hợp (compound proposition)* được xây dựng bằng cách tổ hợp một hoặc nhiều mệnh đề thông qua các *toán tử lôgic (logical operators)*. Ngược lại, *mệnh đề nguyên tử (atomic proposition)* không thể biểu diễn được qua các mệnh đề đơn giản hơn

Phủ định	NOT	$\neg$
Phép hội	AND	$\wedge$
Phép tuyển	OR	$\vee$
Phép tuyển loại	XOR	$\oplus$
Phép kéo theo	IMPLIES	$\rightarrow$
Phép tương đương	IFF	$\leftrightarrow$

- Mỗi quan hệ giữa các giá trị chân lý của các mệnh đề được thể hiện thông qua *bảng chân trị (truth table)*

Lôgic mệnh đề

Toán tử lôgic và bảng chân trị



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

5

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

- **Phủ định (negation)** của mệnh đề p , ký hiệu $\neg p$ hoặc $\bar{p}$, là mệnh đề “không phải là p ”. **Giá trị chân lý $\neg p = T$ khi và chỉ khi $p = F$ và $\neg p = F$ khi và chỉ khi $p = T$**

- Với $p :=$ “2 là số chẵn” thì $\neg p :=$ “2 không là số chẵn”

- Bảng chân trị

p	$\neg p$
T	F
F	T

Lôgic mệnh đề

Toán tử lôgic và bảng chân trị



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

6

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

- **Hội (Conjunction)** của hai mệnh đề p và q , ký hiệu $p \wedge q$ hoặc pq , là mệnh đề “ p và q ”. **Giá trị chân lý $p \wedge q = T$ khi và chỉ khi cả p và q đều nhận giá trị T, và trong các trường hợp còn lại $p \wedge q = F$**

- Với $p :=$ “2 là số chẵn” và $q :=$ “2 là số nguyên tố” thì
 $p \wedge q :=$ “2 là số chẵn và 2 là số nguyên tố”

- **Bảng chân trị**

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Lôgic mệnh đề

Toán tử lôgic và bảng chân trị



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

7

- **Tuyển (Disjunction/Inclusive Or)** của hai mệnh đề p và q , ký hiệu $p \vee q$ hoặc $p + q$, là mệnh đề “ p hoặc q ”. **Giá trị chân lý** $p \vee q = F$ khi và chỉ khi cả p và q đều nhận giá trị F , và trong các trường hợp còn lại $p \vee q = T$

- Với $p :=$ “2 là số chẵn” và $q :=$ “2 là số nguyên tố” thì $p \vee q :=$ “2 là số chẵn hoặc 2 là số nguyên tố”

- **Bảng chân trị**

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Lôgic mệnh đề

Toán tử lôgic và bảng chân trị



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

8

- **Tuyến loại (Exclusive Or)** của hai mệnh đề p và q , ký hiệu $p \oplus q$, là mệnh đề “hoặc p hoặc q ”. **Giá trị chân lý $p \oplus q = T$ khi và chỉ khi chính xác một trong hai mệnh đề p và q nhận giá trị T, và trong các trường hợp còn lại $p \oplus q = F$**

- Với $p :=$ “2 là số chẵn” và $q :=$ “2 là số nguyên tố” thì $p \oplus q :=$ “Hoặc 2 là số chẵn hoặc 2 là số nguyên tố, nhưng không phải cả hai”

- **Bảng chân trị**

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

- **Chú ý:** Khi $p = T$ và $q = T$ thì $p + q = T$ nhưng $p \oplus q = F$

Lôgic mệnh đề

Toán tử lôgic và bảng chân trị



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

9

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

- Mệnh đề *kéo theo (implication)* $p \rightarrow q$, với p, q là hai mệnh đề cho trước, là mệnh đề “nếu p , thì q ”. **Giá trị chân lý** $p \rightarrow q = F$ khi và chỉ khi $p = T$ và $q = F$, và trong mọi trường hợp còn lại $p \rightarrow q = T$

- Ta gọi p là “giả thiết (hypothesis)” và q là “kết luận (conclusion)”. Ta cũng nói “ p là điều kiện đủ (sufficient) cho q ” và “ q là điều kiện cần (necessary) cho p ”

- Với $p :=$ “2 là số chẵn” và $q :=$ “2 là số nguyên tố” thì $p \rightarrow q :=$ “Nếu 2 là số chẵn, thì 2 là số nguyên tố”

- Bảng chân trị

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

- **Chú ý:** Giữa p và q *không nhất thiết có quan hệ nguyên nhân-kết quả*. Ví dụ, $p \rightarrow q = F$ không nhất thiết là “từ p không suy ra q ” mà đơn giản chỉ là $p = T$ và $q = F$

Lôgic mệnh đề

Toán tử lôgic và bảng chân trị



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

- Từ $p \rightarrow q$ ta có thể xây dựng một số mệnh đề mới
 - $q \rightarrow p$ là **mệnh đề đảo (converse)** của $p \rightarrow q$
 - $\neg q \rightarrow \neg p$ là **mệnh đề phản đảo (contrapositive)** của $p \rightarrow q$
 - $\neg p \rightarrow \neg q$ là **mệnh đề nghịch đảo (inverse)** của $p \rightarrow q$
- Ví dụ với $p \rightarrow q :=$ “Nếu 2 là số chẵn, thì 2 là số nguyên tố”
 - $q \rightarrow p :=$ “Nếu 2 là số nguyên tố, thì 2 là số chẵn”
 - $\neg q \rightarrow \neg p :=$ “Nếu 2 không là số nguyên tố, thì 2 không là số chẵn”
 - $\neg p \rightarrow \neg q :=$ “Nếu 2 không là số chẵn, thì 2 không là số nguyên tố”

10

68

Lôgic mệnh đề

Toán tử lôgic và bảng chân trị



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

11

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

- Mệnh đề *tương đương (bi-implication)* $p \leftrightarrow q$, với p, q là hai mệnh đề cho trước, là mệnh đề “ p khi và chỉ khi q ”. **Giá trị chân lý** $p \leftrightarrow q = T$ khi và chỉ khi p và q nhận cùng giá trị, và trong các trường hợp khác $p \leftrightarrow q = F$

- Với $p :=$ “2 là số chẵn” và $q :=$ “2 là số nguyên tố”, ta có $p \leftrightarrow q :=$ “2 là số chẵn khi và chỉ khi 2 là số nguyên tố”

- Bảng chân trị

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Lôgic mệnh đề

Toán tử lôgic và bảng chân trị



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

12

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

■ Tổng kết các toán tử lôgic đã đề cập

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	F	T	T	F	T	T
T	F	F	F	T	T	F	F
F	T	T	F	T	T	T	F
F	F	T	F	F	F	T	T

- Toán tử $\neg$ được gọi là một **toán tử một ngôi (unary operator)**
- Các toán tử $\wedge, \vee, \oplus, \rightarrow, \leftrightarrow$ được gọi là các **toán tử hai ngôi (binary operator)**
- **Thứ tự ưu tiên** của các toán tử lôgic trong một mệnh đề phức hợp: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$. Nên sử dụng **ngoặc đơn** “(” và “)” để xác định thứ tự ưu tiên
 - $\neg p \wedge q$ nghĩa là $(\neg p) \wedge q$ chứ không phải $\neg(p \wedge q)$
 - $p \wedge q \rightarrow r$ nghĩa là $(p \wedge q) \rightarrow r$ chứ không phải $p \wedge (q \rightarrow r)$

Lôgic mệnh đề

Toán tử lôgic và bảng chân trị



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

13

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Bài tập 3

Cho các mệnh đề $p :=$ “Tôi mua một vé xổ số Vietlott” và $q :=$ “Tôi trúng giải đặc biệt 200 tỷ VNĐ”. Hãy mô tả bằng câu thông thường các mệnh đề phức hợp sau:

(a) $\neg p$

(b) $p \vee q$

(c) $p \rightarrow q$

(d) $p \wedge q$

(e) $p \leftrightarrow q$

(f) $\neg p \rightarrow \neg q$

(g) $\neg p \wedge \neg q$

(h) $\neg p \vee (p \wedge q)$

Lôgic mệnh đề

Toán tử lôgic và bảng chân trị



Ví dụ 1

Xây dựng bảng chân trị cho mệnh đề $(p \vee \neg q) \rightarrow q$

p	q	$\neg q$	$p \vee \neg q$	$(p \vee \neg q) \rightarrow q$
T	T	F	T	T
T	F	T	T	F
F	T	F	F	T
F	F	T	T	F

Ví dụ 2

Xây dựng bảng chân trị cho mệnh đề $(p \leftrightarrow q) \leftrightarrow \neg(p \oplus q)$

p	q	$p \leftrightarrow q$	$p \oplus q$	$\neg(p \oplus q)$	$(p \leftrightarrow q) \leftrightarrow \neg(p \oplus q)$
T	T	T	F	T	T
T	F	F	T	F	T
F	T	F	T	F	T
F	F	T	F	T	T

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

14 Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp

chứng minh

Ví dụ và Bài tập

Lôgic mệnh đề

Toán tử lôgic và bảng chân trị



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

15

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Bài tập 4

Xây dựng bảng chân trị cho các mệnh đề $q \rightarrow p$, $\neg q \rightarrow \neg p$, và $\neg p \rightarrow \neg q$

p	q	$\neg p$	$\neg q$	$p \rightarrow q$	$q \rightarrow p$	$\neg q \rightarrow \neg p$	$\neg p \rightarrow \neg q$
T	T			T			
T	F			F			
F	T			T			
F	F			T			

Lôgic mệnh đề

Lôgic và các toán tử bit



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

16

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

- Một **bit** (binary digit = chữ số nhị phân) có giá trị 0 hoặc 1
- Sử dụng bit để biểu diễn giá trị chân lý: 1 cho T và 0 cho F
- Một **chuỗi nhị phân độ dài n** là một dãy sắp thứ tự $x_1x_2 \dots x_n$ trong đó mỗi x_i là một bit ($1 \leq i \leq n$).
 - Ví dụ, 1001101010 là một chuỗi nhị phân độ dài 10
- Các **toán tử bit**: $\neg$ (NOT), $\wedge$ (AND), $\vee$ (OR), $\oplus$ (XOR)

x	y	$\bar{x}$	$x \wedge y$	$x \vee y$	$x \oplus y$
1	1	0	1	1	0
1	0	0	0	1	1
0	1	1	0	1	1
0	0	1	0	0	0

Lôgic mệnh đề

Lôgic và các toán tử bit



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

17

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Tính toán với chuỗi nhị phân: thực hiện theo từng bit

$$\blacksquare \overline{x_1 \dots x_n} = (\overline{x_1}) \dots (\overline{x_n})$$

$$\blacksquare x_1 \dots x_n \wedge y_1 \dots y_n = (x_1 \wedge y_1) \dots (x_n \wedge y_n)$$

$$\blacksquare x_1 \dots x_n \vee y_1 \dots y_n = (x_1 \vee y_1) \dots (x_n \vee y_n)$$

$$\blacksquare x_1 \dots x_n \oplus y_1 \dots y_n = (x_1 \oplus y_1) \dots (x_n \oplus y_n)$$

Bài tập 5

(a) $\overline{11010} =$

(b) $11010 \vee 10001 =$

(c) $11010 \wedge 10001 =$

(d) $11010 \oplus 10001 =$

Các mệnh đề tương đương

Phân loại mệnh đề



Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

18

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

- Một **hằng đúng (tautology)** là một mệnh đề phức hợp luôn luôn đúng với mọi giá trị chân lý của các mệnh đề thành phần
 - Ký hiệu **T**
 - $p \vee \neg p$
- Một **mâu thuẫn (contradiction)** là một mệnh đề phức hợp luôn luôn sai với mọi giá trị chân lý của các mệnh đề thành phần
 - Ký hiệu **F**
 - $p \wedge \neg p$
- Một **tiếp liên (contingency)** là một mệnh đề phức hợp không phải là hằng đúng cũng không phải là mâu thuẫn
 - $(p \vee q) \rightarrow r$

Các mệnh đề tương đương

Tương đương logic



Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

- Mệnh đề phức hợp p *tương đương logic (logically equivalent)* với mệnh đề phức hợp q , ký hiệu $p \equiv q$ hoặc $p \Leftrightarrow q$, khi và chỉ khi mệnh đề $p \leftrightarrow q$ là một hằng đúng
- **Chú ý:** p và q là tương đương logic khi và chỉ khi p và q cùng nhận một giá trị chân lý giống nhau trong mỗi hàng tương ứng của các bảng chân trị của chúng

Các mệnh đề tương đương

Tương đương logic



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Ví dụ 3

Chứng minh rằng $\neg(p \wedge q) \equiv \neg p \vee \neg q$ (luật De Morgan)

p	q	$p \wedge q$	$\neg p$	$\neg q$	$\neg p \vee \neg q$	$\neg(p \wedge q)$
T	T	T	F	F	F	F
T	F	F	F	T	T	T
F	T	F	T	F	T	T
F	F	F	T	T	T	T

Bài tập 6

Chứng minh các tương đương lôgic sau bằng bảng chân trị

(a) $p \oplus q \equiv (p \wedge \neg q) \vee (\neg p \wedge q)$ (b) $p \oplus q \equiv (p \vee q) \wedge \neg(p \wedge q)$

20

68

Các mệnh đề tương đương

Tương đương logic



Một số tương đương logic quan trọng

Tên gọi	Tương đương logic
Luật đồng nhất (Identity laws)	$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$
Luật nuốt (Domination laws)	$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$
Luật lũy đẳng (Idempotent laws)	$p \vee p \equiv p$ $p \wedge p \equiv p$
Luật phủ định kép (Double negation laws)	$\neg(\neg p) \equiv p$
Luật giao hoán (Commutative laws)	$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$
Luật kết hợp (Associative laws)	$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
Luật phân phối (Distributive laws)	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$

Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

21 Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Các mệnh đề tương đương

Tương đương logic



Một số tương đương logic quan trọng (tiếp)

Tên gọi	Tương đương logic
Luật De Morgan (De Morgan's laws)	$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$
Luật hấp thụ (Absorption laws)	$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$
Luật phủ định (Negation laws)	$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$

Chú ý: Trong bảng các tương đương logic quan trọng ở trên, **T** là một mệnh đề phức hợp luôn đúng (hằng đúng) và **F** là một mệnh đề phức hợp luôn sai (mâu thuẫn)

Bài tập 7

Chứng minh các tương đương logic quan trọng nêu trên bằng cách lập bảng chân trị

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

22

Tương đương logic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Các mệnh đề tương đương

Tương đương logic



Ví dụ 4

Chứng minh $\neg(p \vee (\neg p \wedge q))$ và $\neg p \wedge \neg q$ là tương đương logic bằng cách sử dụng các tương đương logic đã biết

$$\neg(p \vee (\neg p \wedge q)) \equiv \neg((p \vee \neg p) \wedge (p \vee q))$$

$$\equiv \neg(\mathbf{T} \wedge (p \vee q))$$

$$\equiv \neg((p \vee q) \wedge \mathbf{T})$$

$$\equiv \neg(p \vee q)$$

$$\equiv \neg p \wedge \neg q$$

Luật phân phối

Luật phủ định

Luật giao hoán

Luật đồng nhất

Luật De Morgan

Bài tập 8

Kiểm tra lại ví dụ trên bằng cách lập bảng chân trị

Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

23

68

Các mệnh đề tương đương

Tương đương logic



Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

24

■ Một số tương đương logic liên quan đến phép kéo theo

$$(1) p \rightarrow q \equiv \neg p \vee q$$

$$(2) p \rightarrow q \equiv \neg q \rightarrow \neg p$$

$$(3) p \vee q \equiv \neg p \rightarrow q$$

$$(4) p \wedge q \equiv \neg(p \rightarrow \neg q)$$

$$(5) \neg(p \rightarrow q) \equiv p \wedge \neg q$$

$$(6) (p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$$

$$(7) (p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$$

$$(8) (p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$$

$$(9) (p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$$

■ Một số tương đương logic liên quan đến phép tương đương

$$(10) p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$$

$$(11) p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$$

$$(12) p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$$

$$(13) \neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$$

Bài tập 9

Chứng minh các tương đương logic trên bằng cách lập bảng chân trị hoặc sử dụng các tương đương logic đã biết

68

Các mệnh đề tương đương

Tương đương logic



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Ví dụ 5

Chứng minh $(p \wedge q) \rightarrow (p \vee q)$ là một hằng đúng

$$(p \wedge q) \rightarrow (p \vee q) \equiv \neg(p \wedge q) \vee (p \vee q)$$

$$\equiv (\neg p \vee \neg q) \vee (p \vee q)$$

$$\equiv (p \vee \neg p) \vee (q \vee \neg q)$$

$$\equiv \mathbf{T} \vee \mathbf{T}$$

$$\equiv \mathbf{T}$$

$$\text{Từ } p \rightarrow q \equiv \neg p \vee q$$

Luật De Morgan

Luật giao hoán, kết hợp

Luật phủ định

Luật nuốt

25

68

Các mệnh đề tương đương

Tương đương logic



Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

- Một tập C các toán tử logic được gọi là **đầy đủ (functionally complete)** nếu mỗi mệnh đề phức hợp tương đương với một mệnh đề phức hợp chỉ sử dụng các toán tử trong C

- $C = \{\neg, \wedge, \vee\}$ là một tập (các toán tử logic) đầy đủ

Ví dụ 6

Tìm một mệnh đề tương đương của $p \leftrightarrow q$ chỉ sử dụng các toán tử $\neg, \wedge, \vee$

- Ứng với **mỗi hàng có giá trị T** ở cột $p \leftrightarrow q$, ta tìm một biểu thức **chỉ đúng với các giá trị của p và q ở hàng đó, và sai với mọi giá trị khác.**

p	q	$p \leftrightarrow q$	
T	T	T	$p \wedge q$
T	F	F	
F	T	F	
F	F	T	$\neg p \wedge \neg q$

- $p \leftrightarrow q$ đúng khi **ít nhất một** biểu thức trên có giá trị T

$$(p \wedge q) \vee (\neg p \wedge \neg q)$$

Chú ý: Phương pháp sử dụng trong ví dụ trên có thể áp dụng với mọi mệnh đề phức hợp. Mệnh đề thu được gọi là **dạng tuyển chuẩn tắc (disjunctive normal form - DNF)**

26

68

Các mệnh đề tương đương

Tương đương logic



- Một tập C các toán tử logic được gọi là **đầy đủ (functionally complete)** nếu mỗi mệnh đề phức hợp tương đương với một mệnh đề phức hợp chỉ sử dụng các toán tử trong C
 - $C = \{\neg, \wedge, \vee\}$ là một tập (các toán tử logic) đầy đủ

Ví dụ 6

Tìm một mệnh đề tương đương của $p \leftrightarrow q$ chỉ sử dụng các toán tử $\neg, \wedge, \vee$

- Ứng với **mỗi hàng có giá trị F** ở cột $p \leftrightarrow q$, ta tìm một biểu thức **chỉ sai với các giá trị của p và q ở hàng đó, và đúng với mọi giá trị khác.**

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

$\neg p \vee q$

$p \vee \neg q$

- $p \leftrightarrow q$ sai khi **tất cả** biểu thức trên có giá trị F

$$(\neg p \vee q) \wedge (p \vee \neg q)$$

Chú ý: Phương pháp sử dụng trong ví dụ trên có thể áp dụng với mọi mệnh đề phức hợp. Mệnh đề thu được gọi là **dạng hội chuẩn tắc (conjunctive normal form - CNF)**

Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

26

68

Các mệnh đề tương đương

Tương đương logic



Tìm CNF/DNF sử dụng các tương đương logic đã biết

■ “Khử” $\oplus$, $\rightarrow$, và $\leftrightarrow$ bằng các tương đương logic đã biết

■ $p \rightarrow q \equiv \neg p \vee q$

■ $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

■ $p \oplus q \equiv (p \wedge \neg q) \vee (\neg p \wedge q)$

■ Giảm “phạm vi” của các dấu phủ định $\neg$ thông qua luật De Morgan và luật phủ định kép

■ Chuyển sang CNF hoặc DNF bằng cách sử dụng các luật phân phối và kết hợp

Ví dụ 7

Tìm CNF của $p \vee q \rightarrow r \wedge s$

$$\begin{aligned} p \vee q \rightarrow r \wedge s &\equiv \neg(p \vee q) \vee (r \wedge s) \\ &\equiv (\neg p \wedge \neg q) \vee (r \wedge s) \\ &\equiv (\neg p \vee (r \wedge s)) \wedge (\neg q \vee (r \wedge s)) \\ &\equiv ((\neg p \vee r) \wedge (\neg p \vee s)) \wedge ((\neg q \vee r) \wedge (\neg q \vee s)) \\ &\equiv (\neg p \vee r) \wedge (\neg p \vee s) \wedge (\neg q \vee r) \wedge (\neg q \vee s) \end{aligned}$$

Khử $\rightarrow$

Luật De Morgan

Luật phân phối

Luật phân phối

Luật kết hợp

Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

27

68

Các mệnh đề tương đương

Tương đương logic



Bài tập 10

Trong rất nhiều ngôn ngữ lập trình, các giá trị chân lý True và False được biểu diễn tương ứng thông qua các số 1 và 0. Ví dụ như, trong Python, cả `0 == False` và `1 == True` đều có giá trị True. Do đó, trên thực tế, chúng ta có thể thực hiện các phép toán số học (cộng, trừ, nhân, chia) với các giá trị chân lý! Thêm vào đó, trong rất nhiều ngôn ngữ lập trình (bao gồm Python), bất kỳ thứ gì khác False (hay nói cách khác, bất kỳ thứ gì khác 0) đều có thể coi là True khi xét các biểu thức liên quan đến điều kiện, ví dụ như `if 2 then X else Y` sẽ chạy và thực hiện X.

Giả sử x và y là các biến Boole trong một ngôn ngữ lập trình mà True và False tương ứng lần lượt với 1 và 0. (Nghĩa là, giá trị của x và y là 0 hoặc 1.) Mỗi đoạn mã sau đây bao gồm một điều kiện dựa trên x , y , và các phép toán số học (cộng, trừ, nhân, chia). Hãy viết lại các điều kiện này sử dụng ngôn ngữ của logic mệnh đề.

- (a) `if x * y ...`
- (b) `if x + y ...`
- (c) `if 2 - x - y ...`
- (d) `if x * (1 - y) ...`
- (e) `if x * (1 - y) + (1 - x) * y ...`

Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

28

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

68

Các mệnh đề tương đương

Tương đương logic



Bài tập 11

Tìm CNF hoặc DNF của các mệnh đề

(a) $p \oplus q$

(c) $\neg p \oplus \neg q$

(b) $p \rightarrow q$

(d) $(p \oplus q) \vee (p \oplus \neg q)$

Bài tập 12

Tập các toán tử logic $\mathcal{C}$ sau có đầy đủ không? Vì sao?

(a) $\mathcal{C} = \{\neg, \wedge\}$

(b) $\mathcal{C} = \{\neg, \vee\}$

(c) $\mathcal{C} = \{\wedge, \vee\}$

Bài tập 13

Cho p, q, r là các mệnh đề nguyên tử. Hãy sử dụng các mệnh đề trên và các toán tử logic $\neg, \wedge, \vee$ để biểu diễn mệnh đề sau

“Ít nhất hai trong ba mệnh đề p, q, r là đúng”

Bài tập 14

Chứng minh các tương đương logic sau

(a) $\neg(p \oplus q) \equiv p \leftrightarrow q$

(b) $\neg p \rightarrow (q \rightarrow r) \equiv q \rightarrow (p \vee r)$

Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

29

68



Vị từ

Một *vị từ (predicate)* là một *hàm mệnh đề (propositional function)* (từ tập các đối tượng đến tập các mệnh đề) mô tả thuộc tính của các đối tượng và mối quan hệ giữa chúng

- Các biến (đối tượng) thường được ký hiệu bởi các chữ cái $x, y, z, \dots$ và có thể được thay thế bằng các giá trị cụ thể từ một *miền xác định (domain)* $\mathcal{D}$ tương ứng cho trước
- Các chữ in hoa $P, Q, R, \dots$ thường được dùng để ký hiệu các hàm mệnh đề (vị từ)
- Với $n \geq 1$, ta gọi $P(x_1, \dots, x_n)$ là *vị từ (n -ngôi) ($(n$ -place predicate) xác định trên miền $\mathcal{D} = D_1 \times \dots \times D_n$ nếu $P(a_1, \dots, a_n)$ là một mệnh đề với bộ $(a_1, \dots, a_n)$ bất kỳ trong $\mathcal{D}$ ($a_1 \in D_1, \dots, a_n \in D_n$)*

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

30

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập



Ví dụ 8

- $P(x) := "x \text{ lớn hơn } 3"$ ($P := "lớn hơn 3"$ và x là một biến) với x là số tự nhiên. $P(x)$ là vị từ xác định trên miền $\mathcal{D} = \mathbb{N}$
- $Q(x, y, z) := "x \text{ cho } y \text{ điểm } z"$ với x, y là tên riêng và z là số tự nhiên. $Q(x, y, z)$ là vị từ xác định trên miền $\mathcal{D} = T \times T \times \mathbb{N}$ trong đó T là tập các tên riêng
- $P(x)$ không phải là mệnh đề nhưng $P(3)$ là mệnh đề. $Q(x, y, z)$ không phải là mệnh đề nhưng $Q(\text{Tý}, \text{Tèo}, 10)$ là mệnh đề

Bài tập 15

$P(x) := "x > 0"$ là vị từ xác định trên miền $\mathcal{D} = \mathbb{Z}$. Tìm giá trị chân lý của các mệnh đề sau

(a) $P(3) \vee P(-1)$

(c) $P(3) \rightarrow P(-1)$

(b) $P(3) \wedge P(-1)$

(d) $P(3) \rightarrow \neg P(-1)$

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

31

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập



Lượng từ

Lượng từ (quantifier) (ví dụ như *tất cả*, *nhiều*, *một số*, *không có*, v.v...) thường được sử dụng với vị từ để định lượng (đếm) các đối tượng (biến) “thỏa mãn” vị từ đó

■ Hai lượng từ quan trọng nhất

Lượng từ	Ký hiệu
với mọi (universal quantifier)	$\forall$
tồn tại (existential quantifier)	$\exists$

- $\forall x P(x)$ nghĩa là “**với mọi** giá trị của x trong miền xác định $\mathcal{D}$, $P(x)$ đúng”
- $\exists x P(x)$ nghĩa là “**tồn tại** giá trị của x trong miền xác định $\mathcal{D}$ (nghĩa là có thể có một hoặc nhiều giá trị thỏa mãn), $P(x)$ đúng”
- $P(x)$ **không phải là mệnh đề** nhưng $\forall x P(x)$ và $\exists x P(x)$ là **mệnh đề**

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

32

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập



- $\forall x P(x)$: **với mọi** giá trị của x trong miền xác định $\mathcal{D}$, $P(x)$ đúng
- $\forall x P(x)$ là
 - **đúng** nếu $P(x)$ đúng với mọi x trong $\mathcal{D}$
 - **sai** nếu $P(x)$ sai với ít nhất một giá trị của x trong $\mathcal{D}$
 - Với $\mathcal{D} = \mathbb{R}$ và $P(x) := "x^2 \geq 0"$, mệnh đề $\forall x P(x)$ đúng
 - Với $\mathcal{D} = \mathbb{R}$ và $P(x) := "x^2 - 1 \geq 0"$, mệnh đề $\forall x P(x)$ sai
- Một **phản ví dụ (counterexample)** của mệnh đề $\forall x P(x)$ là một giá trị x trong miền $\mathcal{D}$ sao cho $P(x)$ sai
- Nếu $\mathcal{D} = \emptyset$ thì mệnh đề $\forall x P(x)$ đúng
- Nếu có thể liệt kê tất cả các phần tử của $\mathcal{D}$, ví dụ như $x_1, x_2, \dots, x_n$, thì $\forall x P(x)$ tương đương lôgic với

$$P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n)$$

Lógica vị từ

Lượng từ “tồn tại”



Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

34

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

■ $\exists x P(x)$: **tồn tại** giá trị của x trong miền xác định $\mathcal{D}$ (nghĩa là có thể có một hoặc nhiều giá trị thỏa mãn), $P(x)$ đúng

■ $\exists x P(x)$

■ **đúng** nếu $P(x)$ đúng với ít nhất một x trong $\mathcal{D}$

■ **sai** nếu $P(x)$ sai với mọi x trong $\mathcal{D}$

■ Với $\mathcal{D} = \mathbb{R}$ và $P(x) := "x^2 = 2"$, mệnh đề $\exists x P(x)$ đúng

■ Với $\mathcal{D} = \mathbb{Z}$ và $P(x) := "x^2 = 2"$, mệnh đề $\exists x P(x)$ sai

■ Nếu $\mathcal{D} = \emptyset$ thì mệnh đề $\exists x P(x)$ sai

■ Nếu có thể liệt kê tất cả các phần tử của $\mathcal{D}$, ví dụ như $x_1, x_2, \dots, x_n$, thì $\exists x P(x)$ tương đương logic với

$$P(x_1) \vee P(x_2) \vee \dots \vee P(x_n)$$



Ví dụ 9

Mô tả câu “Tất cả sinh viên trong lớp này đã học môn Lập trình Java” bằng vị từ và lượng từ

- $J(x) :=$ “ x đã học môn Lập trình Java”
- Nếu $\mathcal{D}$ là tập *các sinh viên trong lớp này* $\forall x J(x)$
- Nếu $\mathcal{D}$ là tập *tất cả mọi người*. Đặt $S(x) :=$ “ x là sinh viên trong lớp này” $\forall x (S(x) \rightarrow J(x))$

Chú ý: Tại sao không phải là $\forall x (S(x) \wedge J(x))$?

Ví dụ 10

Mô tả câu “Một số sinh viên trong lớp này đã học môn Lập trình Java” bằng vị từ và lượng từ

- $J(x) :=$ “ x đã học môn Lập trình Java”
- Nếu $\mathcal{D}$ là tập *các sinh viên trong lớp này* $\exists x J(x)$
- Nếu $\mathcal{D}$ là tập *tất cả mọi người*. Đặt $S(x) :=$ “ x là sinh viên trong lớp này” $\exists x (S(x) \wedge J(x))$

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lôgic vị từ

Vị từ

35

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập



Ví dụ 11

Mô tả câu “**Tồn tại duy nhất** một giá trị của x (trong miền xác định $\mathcal{D}$) sao cho $P(x)$ đúng” (Ta cũng viết “ $\exists!x P(x)$ ”) bằng các lượng từ $\forall$ và $\exists$

$$\blacksquare \exists x P(x) \wedge \forall x_1 \forall x_2 ((P(x_1) \wedge P(x_2)) \rightarrow (x_1 = x_2))$$

$$\blacksquare \exists x (P(x) \wedge \forall y (P(y) \rightarrow (y = x)))$$

Bài tập 16

Giả sử miền xác định của tất cả các biến là $\mathcal{D} = \mathbb{R}$. Các mệnh đề sau là đúng hay sai?

(a) $\forall x (x^2 \neq x)$

(b) $\forall x (x^2 \geq x)$

(c) $\forall x (x^2 \neq 2)$

Bài tập 17

Hãy mô tả định nghĩa số chính phương sau bằng vị từ và lượng từ: “Một số tự nhiên được gọi là số chính phương nếu nó là bình phương của một số tự nhiên nào đó”

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lôgic vị từ

Vị từ

36

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập



Chú ý

Các tương đương lôgic đã biết trong lôgic mệnh đề vẫn đúng trong lôgic vị từ

Ví dụ 12

Giả sử biến x nhận giá trị từ miền $\mathcal{D}$. Ta chứng minh

$\forall x (P(x) \wedge Q(x)) \equiv \forall x (Q(x) \wedge P(x))$. Cụ thể, ta chứng minh hai điều

(1) **Nếu $\forall x (P(x) \wedge Q(x))$ đúng thì $\forall x (Q(x) \wedge P(x))$ đúng**

- Giả sử $\forall x (P(x) \wedge Q(x))$ đúng. Do đó, với mọi $a \in \mathcal{D}$, mệnh đề $P(a) \wedge Q(a)$ đúng. Theo luật giao hoán, $Q(a) \wedge P(a)$ đúng. Suy ra $Q(a) \wedge P(a)$ đúng với mọi $a \in \mathcal{D}$. Do đó, $\forall x (Q(x) \wedge P(x))$ đúng

(2) **Nếu $\forall x (Q(x) \wedge P(x))$ đúng thì $\forall x (P(x) \wedge Q(x))$ đúng**

- Giả sử $\forall x (Q(x) \wedge P(x))$ đúng. Do đó, với mọi $a \in \mathcal{D}$, mệnh đề $Q(a) \wedge P(a)$ đúng. Theo luật giao hoán, $P(a) \wedge Q(a)$ đúng. Suy ra $P(a) \wedge Q(a)$ đúng với mọi $a \in \mathcal{D}$. Do đó, $\forall x (P(x) \wedge Q(x))$ đúng

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

37

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập



Chú ý

Không nên sử dụng bảng chân trị để chứng minh hai mệnh đề có vị từ và lượng từ là tương đương lôgic

- Trong trường hợp miền xác định $\mathcal{D}$ là **vô hạn**, ta không xây dựng được bảng chân trị
- Với phần lớn các phát biểu có lượng từ và vị từ, miền xác định luôn là miền vô hạn
- Tất nhiên, khi miền xác định $\mathcal{D}$ là hữu hạn, có thể sử dụng bảng chân trị
 - Giả sử $\mathcal{D} = \{x_1, \dots, x_n\}$
 - $\forall x P(x) \equiv P(x_1) \wedge \dots \wedge P(x_n)$
 - $\exists x P(x) \equiv P(x_1) \vee \dots \vee P(x_n)$

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

38

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập



Bài tập 18

Giả sử biến x nhận giá trị từ miền $\mathcal{D}$. Chứng minh

- (a) $\forall x (P(x) \vee (Q(x) \wedge R(x))) \equiv \forall x ((P(x) \vee Q(x)) \wedge (P(x) \vee R(x)))$
- (b) $\forall x \neg(P(x) \wedge Q(x)) \equiv \forall x (\neg P(x) \vee \neg Q(x))$

Bài tập 19

Giả sử biến x nhận giá trị từ miền $\mathcal{D}$.

- (a) Chứng minh $\forall x (P(x) \wedge Q(x)) \equiv (\forall x P(x)) \wedge (\forall x Q(x))$
- (b) Các mệnh đề $\forall x (P(x) \vee Q(x))$ và $(\forall x P(x)) \vee (\forall x Q(x))$ có tương đương logic hay không? Vì sao?

Bài tập 20

Giả sử biến x nhận giá trị từ miền $\mathcal{D}$.

- (a) Chứng minh $\exists x (P(x) \vee Q(x)) \equiv (\exists x P(x)) \vee (\exists x Q(x))$
- (b) Các mệnh đề $\exists x (P(x) \wedge Q(x))$ và $(\exists x P(x)) \wedge (\exists x Q(x))$ có tương đương logic hay không? Vì sao?

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lôgic vị từ

Vị từ

39

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập



Ví dụ 13

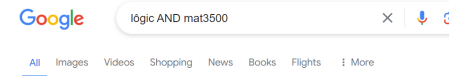
Các *công cụ tìm kiếm (search engine)* như Google, Bing, Yahoo, v.v... *cho phép người dùng sử dụng các toán tử logic (AND, OR, NOT) trong yêu cầu tìm kiếm (query) của họ.*

Ví dụ, yêu cầu (query) *“lógica AND mat3500”*

sẽ trả lại kết quả là các trang web có *chứa đồng thời từ “lógica” và từ “mat3500”*. Nói

một cách đơn giản, *một yêu cầu tìm kiếm có thể được coi như là một vị từ Q* (xác định trên tập hợp tất cả các

trang web có trong cơ sở dữ liệu); công cụ tìm kiếm sẽ *trả lại (theo một thứ tự nào đó) một danh sách các trang web sao cho mệnh đề $Q(p)$ đúng với mỗi trang web p trong danh sách đó.*



Tip: Limit this search to **English** language results. [Learn more about filtering by language](#)

<https://hoanganhduc.github.io> > M... > [Translate this page](#)

VNU-HUS MAT3500 3: Toán rời rạc

Lógica và Chứng minh, slides, bài tập, Phép kéo theo trong logic, Chương 1, 1.1–1.5, 1.7 (Rosen). Các cấu trúc cơ bản I: Tập hợp, Hàm, slides, bài tập, Chương 2 ...

<https://www.studocu.com> > logic-a... > [Translate this page](#)

Logic and Proofs - toán rời rạc - VNU-HUS MAT3500

toán rời rạc mat3500: toán rời rạc bài tập logic và chứng minh hoàng anh đức bộ môn tin học, đại học khtn, đhqg hà nội logic mệnh đề bài tập câu nào sau đây ...

Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

40

68



Bài tập 21

Tìm hiểu một số cách tìm kiếm với Google sử dụng các toán tử lôgic

Bài tập 22 (★)

Xét các yêu cầu tìm kiếm sau:

Q_A : “python AND algorithm AND NOT computer”

Q_B : “(computer OR algorithm) AND python”

Q_C : “python AND NOT (computer OR algorithm OR program)”

Hãy mô tả hoặc lấy ví dụ (một trang web, một câu nào đó, v.v...) về kết quả trả lại trong các trường hợp sau:

- (a) Kết quả trả lại thỏa mãn yêu cầu Q_A
- (b) Kết quả trả lại thỏa mãn yêu cầu Q_A và không thỏa mãn yêu cầu Q_B
- (c) Kết quả trả lại thỏa mãn yêu cầu Q_A và Q_B nhưng không thỏa mãn yêu cầu Q_C

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập



Ví dụ 14

- Ở mức độ nào đó, chúng ta có thể coi một **cơ sở dữ liệu (database)** như là một bảng
 - Các hàng tương ứng với các **cá thể (individual entity)** nào đó
 - Các cột tương ứng với các **trường (field)** mô tả dữ liệu liên quan đến các cá thể
- Một **truy vấn cơ sở dữ liệu (database query)** có thể xem như là một vị từ $Q(x)$ có chứa các điều kiện để kiểm tra các giá trị từ các cột và các toán tử logic liên kết các điều kiện này.
- Khi một truy vấn cơ sở dữ liệu được đưa vào, **hệ quản trị cơ sở dữ liệu (database management system)** sẽ trả lại một danh sách các hàng (ứng với các thực thể) trong cơ sở dữ liệu thỏa mãn điều kiện đề ra trong truy vấn
- Chúng ta có thể nghĩ về hình thức truy cập cơ sở dữ liệu này **từ góc nhìn của logic vị từ: để phản hồi truy vấn (query) Q , hệ thống trả lại một danh sách các hàng, trong đó mỗi hàng x thỏa mãn điều kiện $Q(x)$ đúng**

Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Logic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

42

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập



Bảng: Ví dụ một cơ sở dữ liệu. Nếu muốn tìm danh sách tất cả các sinh viên có GPA (grade point averages - điểm trung bình) tối thiểu 3.4 và nếu đã học ít nhất một khóa học về khoa học máy tính (computer science - CS) thì phải đến từ Hawaii, ta có thể truy vấn $[GPA(x) \geq 3.4] \wedge [takenCS(x) \rightarrow (home(x) = Hawaii)]$. Kết quả trả lại với cơ sở dữ liệu này là Charlie.

name	GPA	CS taken?	home	age
Alice	4.0	yes	Alberta	20
Bob	3.14	yes	Bermuda	19
Charlie	3.54	no	Cornwall	18
Desdemona	3.8	yes	Delaware	17



- Trước đó, ta thường phải chỉ rõ miền xác định $\mathcal{D}$ có chứa các giá trị của biến trước khi phát biểu mệnh đề với vị từ và lượng từ. Để thuận tiện, ***có thể chỉ ra $\mathcal{D}$ ngay trong mệnh đề***

- $\forall x > 0 P(x)$ nghĩa là “Với mọi số $x > 0$, $P(x)$ đúng”. ($\mathcal{D}$ là tập tất cả các số lớn hơn không.) Thực ra, đây là cách viết ngắn gọn của mệnh đề $\forall x Q(x)$ trong đó

$$Q(x) := (x > 0) \rightarrow P(x)$$

- $\exists x > 0 P(x)$ nghĩa là “Tồn tại số $x > 0$, $P(x)$ đúng”. ($\mathcal{D}$ là tập tất cả các số lớn hơn không.) Thực ra, đây là cách viết ngắn gọn của mệnh đề $\exists x Q(x)$ trong đó

$$Q(x) := (x > 0) \wedge P(x)$$

- Chú ý:

$$\forall x > 0 P(x) \neq \forall x (x > 0 \wedge P(x))$$

$$\exists x > 0 P(x) \neq \exists x (x > 0 \rightarrow P(x))$$

- Các lượng từ $\forall$ và $\exists$ có ***thứ tự ưu tiên cao hơn tất cả các toán tử lôgic*** đã đề cập

- $\forall x P(x) \vee Q(x)$ nghĩa là $(\forall x P(x)) \vee Q(x)$ chứ không phải $\forall x (P(x) \vee Q(x))$

Lôgic vị từ

Biến tự do và biến ràng buộc



- Vị từ $P(x)$ có **biến tự do (free variable)** x (nghĩa là, giá trị của x không xác định)
- Lượng từ ($\forall$ hoặc $\exists$) sử dụng với một vị từ có một hoặc nhiều biến tự do “ràng buộc” những biến này, tạo thành một biểu thức có một hoặc nhiều **biến ràng buộc (bound variable)**

Ví dụ 15

- $P(x, y)$ có hai biến tự do: x và y
- $\forall x P(x, y)$ có một biến tự do y và một biến ràng buộc x
- Biểu thức **không có bất kỳ biến tự do nào**, ví dụ $\forall x P(x)$, là mệnh đề
- Biểu thức **có một hoặc nhiều biến tự do**, ví dụ $\forall x P(x, y)$, không là mệnh đề

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

45

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Lôgic vị từ

Phủ định với lượng từ



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

■ *Phủ định* của mệnh đề có lượng từ

$$\blacksquare \neg \forall x P(x) \equiv \exists x \neg P(x)$$

$$\blacksquare \neg \exists x P(x) \equiv \forall x \neg P(x)$$

■ Hai tương đương lôgic trên được gọi là *Luật De Morgan cho lượng từ (De Morgan's Laws for Quantifiers)*. Lý do của tên gọi này là nếu ta có thể liệt kê toàn bộ các phần tử trong miền $\mathcal{D}$, ví dụ như $x_1, \dots, x_n$, thì

$$\begin{aligned}\neg \forall x P(x) &\equiv \neg(P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n)) \\ &\equiv \neg P(x_1) \vee \neg P(x_2) \vee \dots \vee \neg P(x_n) \\ &\equiv \exists x \neg P(x)\end{aligned}$$

Luật De Morgan

$$\begin{aligned}\neg \exists x P(x) &\equiv \neg(P(x_1) \vee P(x_2) \vee \dots \vee P(x_n)) \\ &\equiv \neg P(x_1) \wedge \neg P(x_2) \wedge \dots \wedge \neg P(x_n) \\ &\equiv \forall x \neg P(x)\end{aligned}$$

Luật De Morgan

46

68



Bài tập 23

Giả sử A là một mệnh đề thỏa mãn điều kiện A đúng hay sai không phụ thuộc vào x . Giả sử miền xác định của các biến là không rỗng. Hãy chứng minh các tương đương lôgic sau

(a) $(\forall x P(x)) \vee A \equiv \forall x (P(x) \vee A)$

(b) $(\exists x P(x)) \vee A \equiv \exists x (P(x) \vee A)$

(c) $(\forall x P(x)) \wedge A \equiv \forall x (P(x) \wedge A)$

(d) $(\exists x P(x)) \wedge A \equiv \exists x (P(x) \wedge A)$

(e) $A \rightarrow \forall x P(x) \equiv \forall x (A \rightarrow P(x))$

(f) $\exists x P(x) \rightarrow A \equiv \forall x (P(x) \rightarrow A)$

(g) $A \rightarrow \exists x P(x) \equiv \exists x (A \rightarrow P(x))$

(h) $\forall x P(x) \rightarrow A \equiv \exists x (P(x) \rightarrow A)$

(Gợi ý: Xét từng trường hợp $A = T$ và $A = F$)

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

47

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập



Ví dụ 16

Ta chứng minh câu (h) trong Bài tập 23:

$\forall x P(x) \rightarrow A \equiv \exists x (P(x) \rightarrow A)$, trong đó A là mệnh đề có giá trị chân lý không phụ thuộc vào biến x

$$\begin{aligned}\exists x (P(x) \rightarrow A) &\equiv \exists x (\neg P(x) \vee A) \\ &\equiv \exists x \neg P(x) \vee \exists x A \\ &\equiv \exists x \neg P(x) \vee A \\ &\equiv \neg \forall x P(x) \vee A \\ &\equiv \forall x P(x) \rightarrow A\end{aligned}$$

$$p \rightarrow q \equiv \neg p \vee q$$

Bài 20(a)

Định nghĩa của A

De Morgan cho lượng từ

$$p \rightarrow q \equiv \neg p \vee q$$

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

48

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập



Ví dụ 17

$P(x, y) :=$ “ x nhỏ hơn y ” xác định trên miền $\mathcal{D} = \mathbb{Z} \times \mathbb{Z}$

- $\exists y P(x, y) :=$ “có số nguyên y sao cho x nhỏ hơn y ” (Biểu thức với 1 biến tự do—không phải mệnh đề)
- $\forall x (\exists y P(x, y)) :=$ “với mọi số nguyên x có số nguyên y sao cho x nhỏ hơn y ” (Biểu thức với 0 biến tự do—là mệnh đề)

Bài tập 24

Cho $x \in \mathbb{Z}$ và $y \in \mathbb{Z}$ và $P(x, y) := “x < y”$. Xác định giá trị chân lý của các mệnh đề sau

- | | |
|-----------------------------------|-----------------------------------|
| (a) $\forall x \forall y P(x, y)$ | (c) $\exists x \forall y P(x, y)$ |
| (b) $\forall x \exists y P(x, y)$ | (d) $\exists x \exists y P(x, y)$ |



■ Một số tương đương lôgic:

■ $\forall x \forall y P(x, y) \equiv \forall y \forall x P(x, y)$

■ $\exists x \exists y P(x, y) \equiv \exists y \exists x P(x, y)$

■ Để thuận tiện, có thể nối các lượng từ cùng loại

■ $\forall x \forall y P(x, y) \equiv \forall x, y P(x, y)$

■ Trừ khi tất cả các lượng từ đều là $\forall$ hoặc đều là $\exists$, *thứ tự các lượng từ là quan trọng*

■ $\forall x \exists y P(x, y)$ khác với $\exists y \forall x P(x, y)$

■ Ví dụ, với x, y là các số nguyên, mệnh đề $\forall x \exists y (x < y)$ đúng, vì với mỗi x ta có thể chọn $y = x + 1$ và hiển nhiên $x < y$. Ngược lại, mệnh đề $\exists y \forall x (x < y)$ sai, vì không tồn tại số nguyên lớn nhất



Bài tập 25

Các mệnh đề sau khi nào đúng và khi nào sai?

(1) $\forall x \forall y P(x, y) \equiv \forall y \forall x P(x, y)$

(2) $\forall x \exists y P(x, y)$

(3) $\exists y \forall x P(x, y)$

(4) $\exists x \exists y P(x, y) \equiv \exists y \exists x P(x, y)$

Bài tập 26

Hãy biểu diễn phủ định của các mệnh đề sau sao cho tất cả các ký tự phủ định $\neg$ đứng ngay trước các vị từ

(a) $\forall x \exists y \forall z T(x, y, z)$

(b) $\forall x \exists y P(x, y) \vee \forall x \exists y Q(x, y)$

(c) $\forall x \exists y (P(x, y) \wedge \exists z R(x, y, z))$

(d) $\forall x \exists y (P(x, y) \rightarrow Q(x, y))$

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

51

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Chứng minh

Một số thuật ngữ



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

52

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

- **Chứng minh (proof)**: một lý luận hợp lý chỉ ra tính đúng đắn của một mệnh đề toán học.
- **Tiên đề (axiom/postulate)**: một mệnh đề được giả thiết là đúng
- **Định lý (theorem)**: một mệnh đề đã được chứng minh là đúng
- **Mệnh đề (proposition)**: một định lý “không quá quan trọng”
- **Bổ đề (lemma)**: một định lý nhỏ có thể được sử dụng như một công cụ hỗ trợ chứng minh các định lý khác lớn hơn
- **Hệ quả (corollary)**: một định lý nhỏ thu được bằng cách trực tiếp áp dụng một định lý khác lớn hơn
- **Giả thuyết (conjecture)**: một mệnh đề mà tính đúng/sai của nó chưa được xác định, nhưng thường được “tin là đúng” thông qua một số bằng chứng hoặc qua kinh nghiệm, dự đoán của một chuyên gia

Chứng minh

Một số phương pháp chứng minh



Lógica và Chứng minh

Hoàng Anh Đức

Mục tiêu

Chứng minh p đúng

- **Chứng minh trực tiếp (direct proof)**
- **Chứng minh gián tiếp (indirect proof):** Giả thiết $\neg p$ đúng, chứng minh $\neg p \rightarrow \mathbf{F}$ (phương pháp **Chứng minh phản chứng (Proof by Contradiction)**)
- **Chứng minh bằng cách chia trường hợp (proof by cases):** Tìm $p_1, \dots, p_n$ thỏa mãn $p \equiv p_1 \wedge p_2 \wedge \dots \wedge p_n$ và chứng minh p_i đúng với mỗi $i \in \{1, 2, \dots, n\}$. (Chia điều cần chứng minh p thành các phần nhỏ (trường hợp) $p_1, \dots, p_n$ và xét từng phần một cách riêng biệt)

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

53

Một số phương pháp chứng minh

Ví dụ và Bài tập

Chứng minh

Một số phương pháp chứng minh



Mục tiêu

Chứng minh $p \rightarrow q$ đúng

- **Chứng minh hiển nhiên (trivial proof):** Chứng minh q đúng mà không cần giả thiết gì khác
- **Chứng minh trực tiếp (direct proof):** Giả thiết p đúng, chứng minh q
- **Chứng minh gián tiếp (indirect proof)**
 - **Chứng minh phản đảo (Proof by Contraposition)**
($\neg q \rightarrow \neg p$): Giả thiết $\neg q$ đúng, chứng minh $\neg p$
 - **Chứng minh phản chứng (Proof by Contradiction):** Giả thiết $p \wedge \neg q$ đúng, và chỉ ra rằng điều này dẫn đến một mâu thuẫn (nghĩa là, chứng minh $(p \wedge \neg q) \rightarrow \text{F}$)
- **Chứng minh rỗng (vacuous proof):** Chứng minh $\neg p$ đúng mà không cần giả thiết gì khác
- **Chứng minh bằng cách chia trường hợp (proof by cases):** Tìm $p_1, p_2, \dots, p_n$ thỏa mãn $p \rightarrow q \equiv (p_1 \rightarrow q) \wedge (p_2 \rightarrow q) \wedge \dots \wedge (p_n \rightarrow q)$ và chứng minh $p_i \rightarrow q$ đúng với từng giá trị $i \in \{1, 2, \dots, n\}$

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

54

68

Chứng minh

Một số phương pháp chứng minh



Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

55

Một số phương pháp chứng minh

Ví dụ và Bài tập

Mục tiêu

Chứng minh $p \leftrightarrow q$ đúng

- $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
- Chứng minh $p \rightarrow q$ đúng và $q \rightarrow p$ đúng

Chứng minh

Ví dụ



Một số nguyên n là số **chẵn** (*even*) khi và chỉ khi $n = 2k$ với k là số nguyên nào đó; n là số **lẻ** (*odd*) khi và chỉ khi $n = 2k + 1$ với k là số nguyên nào đó

Định lý 1

Với mọi số nguyên n , n không thể vừa chẵn vừa lẻ

Chứng minh phản chứng.

- (1) **Nhắc lại:** Để chứng minh p , ta chứng minh $\neg p \rightarrow \mathbf{F}$
- (2) Giả sử tồn tại một số nguyên n vừa chẵn vừa lẻ. Ta chứng minh tồn tại mâu thuẫn
- (3) Do n chẵn, $n = 2k$ với số nguyên k nào đó
- (4) Do n lẻ, $n = 2j + 1$ với số nguyên j nào đó
- (5) Do đó, $2k = 2j + 1$, suy ra $k - j = \frac{1}{2}$. Mệnh đề này sai với mọi số nguyên k và j , đây là một mâu thuẫn. Ta có điều phải chứng minh

Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

56

68

Chứng minh

Ví dụ



Chứng minh sau của Định lý 1

Với mọi số nguyên n , n không thể vừa chẵn vừa lẻ

là sai? Tại sao?

Chứng minh phản chứng.

- (1) Giả sử tồn tại một số nguyên n vừa chẵn vừa lẻ. Ta chứng minh tồn tại mâu thuẫn
- (2) Do n chẵn, $n = 2k$ với số nguyên k nào đó
- (3) Do n lẻ, $n = 2k + 1$ với số nguyên k nào đó
- (4) Do đó, $2k = 2k + 1$, suy ra $0 = 1$. Mệnh đề này sai với mọi số nguyên k , đây là một mâu thuẫn. Ta có điều phải chứng minh



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

57

68

Chứng minh

Bài tập



Bài tập 27

Chứng minh bằng phương pháp phản chứng rằng

- (a) Tổng của một số vô tỷ và một số hữu tỷ là một số vô tỷ
- (b) Với mọi số nguyên n , nếu $n^3 + 5$ lẻ, thì n chẵn
- (c) Với mọi số nguyên n , nếu $3n + 2$ chẵn, thì n chẵn

Bài tập 28

Chứng minh bằng phương pháp phản chứng rằng

- (a) Có ít nhất mười ngày trong 64 ngày bất kỳ rơi vào cùng một ngày của một tuần (nghĩa là, có ít nhất mười ngày cùng là Thứ Hai, hoặc cùng là Thứ Ba, v.v...)
- (b) Có ít nhất ba ngày trong 25 ngày bất kỳ rơi vào cùng một tháng của năm

Bài tập 29

Sử dụng phương pháp phản chứng, hãy chứng minh rằng $\sqrt{2}$ không phải là một số hữu tỷ

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

58

68

Chứng minh

Ví dụ



Định lý 2

Với mọi số nguyên n , nếu n là số lẻ, thì n^2 cũng là số lẻ

Chứng minh trực tiếp.

- (1) Nếu n lẻ, thì $n = 2k + 1$ với k là số nguyên nào đó
- (2) Do đó, $n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1$
- (3) Do đó, $n^2 = 2j + 1$ với $j = 2k^2 + 2k$ là số nguyên
- (4) Theo định nghĩa, n^2 lẻ



Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

59

68



Bài tập 30

Chứng minh trực tiếp các mệnh đề sau

- (a) Tổng của hai số lẻ là một số chẵn
- (b) Tổng của hai số chẵn là một số chẵn
- (c) Bình phương của một số chẵn là một số chẵn
- (d) Tích của hai số lẻ là một số lẻ

Chứng minh

Ví dụ



Định lý 3

Với mọi số nguyên n , nếu $3n + 2$ là số lẻ, thì n cũng là số lẻ

Chứng minh phản đảo.

- **Nhắc lại:** để chứng minh $p \rightarrow q$, ta chứng minh $\neg q \rightarrow \neg p$
 - *Ta chứng minh mệnh đề phản đảo: Với mọi số nguyên n , nếu n chẵn, thì $3n + 2$ cũng chẵn*
- (1) Giả sử n chẵn
 - (2) Do đó $n = 2k$ với số nguyên k nào đó
 - (3) Suy ra $3n + 2 = 3(2k) + 2 = 6k + 2 = 2(3k + 1)$
 - (4) Từ đó, $3n + 2 = 2j$ với $j = 3k + 1$ là số nguyên. Do đó $3n + 2$ chẵn



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

61

68

Chứng minh

Ví dụ



Định lý 4

Với mọi số nguyên n , $n(n+1)^2$ là một số chẵn

Chứng minh bằng cách chia trường hợp.

- **Nếu n chẵn**, thì $n = 2k$ với số nguyên k nào đó. Suy ra $n(n+1)^2 = 2j$ với $j = k(2k+1)^2$ là số nguyên. Do đó, $n(n+1)^2$ là một số chẵn
- **Nếu n lẻ**, thì $n = 2k+1$ với số nguyên k nào đó. Suy ra $n(n+1)^2 = 2\ell$ với $\ell = (2k+1)2(k+1)^2$ là số nguyên. Do đó, $n(n+1)^2$ là một số chẵn



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

62

68

Chứng minh

Ví dụ



Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

Chứng minh sau của mệnh đề

(Với mọi số nguyên n) Ta có $-|n| \leq n \leq |n|$

là sai. Tại sao?

Chứng minh bằng cách chia trường hợp.

- (1) Nếu $n > 0$, ta có $-n \leq 0 \leq n$. Theo định nghĩa, $|n| = n$ và do đó $-|n| = -n$. Suy ra $-|n| = -n \leq 0 \leq n = |n|$
- (2) Nếu $n < 0$, ta có $n \leq 0 \leq -n$. Theo định nghĩa, $|n| = -n$ và do đó $-|n| = n$. Suy ra $-|n| = n \leq 0 \leq -n = |n|$



63

68

Chứng minh

Ví dụ



Định lý 5

Với mọi số nguyên n , nếu n vừa chẵn vừa lẻ, thì $n^2 = n + n$

Chứng minh rằng.

- **Nhắc lại:** Để chứng minh $p \rightarrow q$, ta chứng minh $\neg p$ mà không cần bất cứ giả thiết nào
 - Để chứng minh mệnh đề đã cho bằng cách chứng minh rằng, ta cần chứng minh với mọi số nguyên n , mệnh đề “ n vừa chẵn vừa lẻ” là sai
- (1) Định lý 1: Mệnh đề “ n vừa chẵn vừa lẻ” sai với mọi số nguyên n
 - (2) Ta có điều phải chứng minh (Tập các giả thiết là rỗng)



Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

64

68

Chứng minh

Ví dụ



Định lý 6

Với mọi số nguyên n , nếu n là tổng của hai số nguyên tố, thì hoặc n chẵn hoặc n lẻ

Chứng minh hiển nhiên.

- **Nhắc lại:** Để chứng minh $p \rightarrow q$, ta chứng minh q mà không cần bất cứ giả thiết nào
 - Để chứng minh mệnh đề đã cho bằng cách chứng minh hiển nhiên, ta cần chứng minh với mọi số nguyên n , mệnh đề “hoặc n chẵn hoặc n lẻ” là đúng
- (1) Định lý 1: Với mọi số nguyên n , mệnh đề “hoặc n chẵn hoặc n lẻ” đúng
 - (2) Do đó, mệnh đề cần chứng minh luôn đúng

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

65



68

Chứng minh

Ví dụ



Chứng minh sau của mệnh đề

$$1 = 2$$

là sai. Tại sao?

Chứng minh.

Gọi a, b là hai số nguyên dương bằng nhau.

(1) $a = b$

(2) $a^2 = ab$

(3) $a^2 - b^2 = ab - b^2$

(4) $(a - b)(a + b) = (a - b)b$
nhân tử

(5) $a + b = b$

(6) $2b = b$

(7) $2 = 1$

Giả thiết

Nhân hai vế của (1) với a

Trừ b^2 từ cả hai vế của (2)

Phân tích hai vế của (3) thành

Chia cả hai vế của (4) cho $a - b$

Thay a bởi b trong (5) vì $a = b$, và đơn giản hóa

Chia cả hai vế của (6) cho b

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

66

68

Chứng minh

Ví dụ



Chứng minh sau của mệnh đề

(Với mọi số nguyên n) Nếu n^2 chẵn, thì n cũng chẵn

là sai. Tại sao?

Chứng minh.

- (1) Mệnh đề đúng với $n = 0$. Do đó ta chỉ xét $n \neq 0$
- (2) Giả sử n^2 chẵn. Do đó $n^2 = 2k$ với số nguyên k nào đó
- (3) Chia cả hai vế cho n , ta có $n = (2k)/n = 2(k/n)$
- (4) Do đó, tồn tại số $j = k/n$ sao cho $n = 2j$
- (5) Do tích của j và một số nguyên (2) là một số nguyên (n), nên j cũng là số nguyên
- (6) Do đó n chẵn

Lógica và Chứng minh

Hoàng Anh Đức

Lógica mệnh đề

Mệnh đề

Toán tử logic và bảng chân trị

Lógica và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương logic

Lógica vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lồng các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

67



68



Bài tập 31

Chứng minh các mệnh đề sau. Nêu rõ phương pháp chứng minh bạn sử dụng

- (a) Với mọi số nguyên n , nếu n chẵn thì $(-1)^n = 1$
- (b) Với mọi số nguyên x, y, z , nếu $x + y + z$ lẻ, thì ít nhất một trong ba số x, y, z là lẻ
- (c) Với mọi số nguyên m và n , nếu $m \cdot n$ chẵn, thì m chẵn hoặc n chẵn
- (d) Với mọi số nguyên dương n , n chẵn khi và chỉ khi $7n + 4$ chẵn
- (e) Với mọi số nguyên dương n , n lẻ khi và chỉ khi $5n + 6$ lẻ
- (f) Với mọi số nguyên dương m và n , $m^2 = n^2$ khi và chỉ khi $m = n$ hoặc $m = -n$

Lôgic và Chứng minh

Hoàng Anh Đức

Lôgic mệnh đề

Mệnh đề

Toán tử lôgic và bảng chân trị

Lôgic và các toán tử bit

Các mệnh đề tương đương

Phân loại mệnh đề

Tương đương lôgic

Lôgic vị từ

Vị từ

Lượng từ

Phủ định với lượng từ

Lỗi các lượng từ

Chứng minh

Một số thuật ngữ

Một số phương pháp chứng minh

Ví dụ và Bài tập

68

68

Part I

Phụ lục

Các quy tắc suy luận

Giới thiệu



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

- **Một lập luận (argument)** là một dãy các phát biểu kết thúc bằng một kết luận
- Một số dạng lập luận (“hợp lý (valid)”) không bao giờ dẫn tới một kết luận sai từ các phát biểu đúng. Một số dạng lập luận khác (“ngụy biện (fallacies)”) có thể dẫn tới một kết luận sai từ các phát biểu đúng
- **Một lập luận lôgic (logical argument)** bao gồm một dãy các mệnh đề (có thể là mệnh đề phức hợp) được gọi là **các tiền đề (premises)**/**các giả thiết (hypotheses)** và một mệnh đề duy nhất gọi là **kết luận (conclusion)**
- **Các quy tắc suy luận lôgic (logical rules of inference)** là các phương pháp chỉ phụ thuộc vào lôgic để suy ra một phát biểu mới từ một tập hợp các phát biểu khác. (Có thể coi các quy tắc này như là các mẫu (templates) cho việc xây dựng các lập luận hợp lý (valid arguments))

Các quy tắc suy luận

Giới thiệu



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

2

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số quy tắc suy luận phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

■ Một lập luận logic được gọi là **hợp lý (valid)** nếu **khi mọi giả thiết đúng thì kết luận cũng đúng**. Một lập luận logic không hợp lý được gọi là một **ngụy biện (fallacy)**

■ Một **quy tắc suy luận (inference rule)** là một khuôn mẫu thiết lập rằng nếu chúng ta **biết một tập các giả thiết nào đó là đúng**, thì chúng ta có thể **suy luận một cách hợp lý rằng một phát biểu kết luận liên quan nào đó là đúng**

■ Mỗi **quy tắc suy luận logic hợp lý (valid inference rule)** tương ứng với một **hằng đúng (tautology)**

■ Ký hiệu $\therefore$ nghĩa là “do đó”

■ Hằng đúng (tautology) tương ứng (nếu quy tắc hợp lý)

Giả thiết 1
Giả thiết 2

$\therefore$ $\frac{\dots}{\text{Kết luận}}$

$((\text{Giả thiết 1}) \wedge (\text{Giả thiết 2}) \wedge \dots) \rightarrow (\text{Kết luận})$

Các quy tắc suy luận

Giới thiệu



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

3

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên biện phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Ví dụ 18

- Một lập luận logic
 - **Giả thiết 1:** Nếu tôi làm việc suốt đêm, thì tôi mệt mỗi
 - **Giả thiết 2:** Tôi làm việc suốt đêm
 - **Kết luận:** Do đó, tôi mệt mỗi
- Biểu diễn các biến logic
 - p = “Tôi làm việc suốt đêm”
 - q = “Tôi mệt mỗi”
- Theo góc nhìn của logic, lập luận trên có thể được viết lại như sau:

$$\begin{array}{ll} p \rightarrow q & \text{Giả thiết 1} \\ p & \text{Giả thiết 2} \\ \hline \therefore q & \text{Kết luận} \end{array}$$

Các quy tắc suy luận

Một số quy tắc suy luận trong logic mệnh đề



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên biện phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

- Quy tắc *Modus Ponens* (tiếng Latin của “phương pháp khẳng định (method of affirming)”)

$$\begin{array}{c} p \\ p \rightarrow q \\ \hline \therefore q \end{array}$$

Modus Ponens

$$(p \wedge (p \rightarrow q)) \rightarrow q$$

Hằng đúng tương ứng

- Chú ý rằng hàng đầu tiên là hàng duy nhất mà giả thiết luôn đúng

Ví dụ 19

Nếu $\left\{ \begin{array}{l} p \rightarrow q \\ p \end{array} \right.$ “Nếu n chia hết cho 3, thì n^2 chia hết cho 3” } là ĐÚNG
thì $\therefore q$ “ n chia hết cho 3” } cũng ĐÚNG
 $\therefore q$ “ n^2 chia hết cho 3”

Các quy tắc suy luận

Một số quy tắc suy luận trong logic mệnh đề



Lôgic và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên biện phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

- Quy tắc **Modus Tollens** (tiếng Latin của “phương pháp phủ định (method of denying)”)

$$\begin{array}{c} \neg q \\ p \rightarrow q \\ \hline \therefore \neg p \end{array}$$

Modus Tollens

$$(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$$

Hằng đúng tương ứng

Ví dụ 20

Nếu $\left\{ \begin{array}{l} p \rightarrow q \text{ “Nếu chiếc nhẫn làm bằng kim cương, thì nó có thể làm xước tấm kính”} \\ \neg q \text{ “Chiếc nhẫn không làm xước tấm kính”} \end{array} \right\}$ là ĐÚNG

thì $\therefore \neg p$ “Chiếc nhẫn không làm bằng kim cương” cũng ĐÚNG

Các quy tắc suy luận

Một số quy tắc suy luận trong logic mệnh đề



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

■ Quy tắc Cộng (Addition)

$$\therefore \frac{p}{p \vee q}$$

Quy tắc Cộng (Addition)

■ Quy tắc Rút gọn (Simplification)

$$\therefore \frac{p \wedge q}{p}$$

Quy tắc Rút gọn
(Simplification)

■ Quy tắc Hội (Conjunction)

$$\therefore \frac{p}{q} \quad \frac{q}{p \wedge q}$$

Quy tắc Hội (Conjunction)

$$p \rightarrow (p \vee q)$$

Hằng đúng tương ứng

$$(p \wedge q) \rightarrow p$$

Hằng đúng tương ứng

$$((p) \wedge (q)) \rightarrow p \wedge q$$

Hằng đúng tương ứng

6

36

Các quy tắc suy luận

Một số quy tắc suy luận trong logic mệnh đề



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

7

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Ví dụ 21

Các lập luận sau sử dụng các quy tắc suy luận nào?

- (a) Nếu có ca nhiễm COVID-19 mới, thì trường sẽ đóng cửa. Trường không đóng cửa hôm nay. Do đó, không có ca nhiễm COVID-19 mới hôm nay [Quy tắc Modus Tollens]
- (b) Nhiệt độ hiện tại là dưới $0^{\circ}C$. Do đó, nhiệt độ hiện tại là dưới $0^{\circ}C$ hoặc trời đang mưa [Quy tắc Cộng]
- (c) Nhiệt độ hiện tại là dưới $0^{\circ}C$ và trời đang mưa. Do đó, nhiệt độ hiện tại là dưới $0^{\circ}C$ [Quy tắc Rút gọn]

Các quy tắc suy luận

Một số quy tắc suy luận trong logic mệnh đề



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

8

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

■ Quy tắc *Tam đoạn luận giả định (Hypothetical syllogism)*

$$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$$

Tam đoạn luận giả định
(Hypothetical syllogism)

$$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$$

Hằng đúng tương ứng

Ví dụ 22

p “Nếu trời mưa hôm nay, thì chúng ta sẽ không tổ chức tiệc nướng hôm nay. Nếu chúng ta không tổ chức tiệc nướng hôm nay, thì chúng ta sẽ tổ chức tiệc nướng vào ngày mai. Do đó nếu trời mưa hôm nay, thì chúng ta sẽ tổ chức tiệc nướng vào ngày mai.” q

r p r

Các quy tắc suy luận

Một số quy tắc suy luận trong logic mệnh đề



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

9

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

■ Quy tắc *Tam đoạn luận tuyển (Disjunctive syllogism)*

$$\begin{array}{c} p \vee q \\ \neg p \\ \hline \therefore q \end{array}$$

Tam đoạn luận tuyển
(Disjunctive syllogism)

$$((p \vee q) \wedge (\neg p)) \rightarrow q$$

Hằng đúng tương ứng

Ví dụ 23

“Ví của tôi nằm trong túi áo khoác hoặc nó nằm trên bàn.
Ví của tôi không nằm trong túi áo khoác. Do đó, nó nằm trên bàn.”

p

q

$\neg p$

q

Các quy tắc suy luận

Một số quy tắc suy luận trong logic mệnh đề



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

■ Quy tắc *Hợp giải (Resolution)*

$$\frac{p \vee q \quad \neg p \vee r}{\therefore q \vee r}$$

Hợp giải (Resolution)

■ Khi $q = r$, ta có

$$((p \vee q) \wedge (\neg p \vee q)) \rightarrow q$$

■ Khi $r = \mathbf{F}$, ta có *Quy tắc tam đoạn luận tuyến*

$$((p \vee q) \wedge (\neg p)) \rightarrow q$$

Ví dụ 24

q “Tôi đang đi trên đường hoặc p trời mưa. $\neg p$ Trời không mưa hoặc
 r tôi đang ở nhà. Do đó, tôi đang đi trên đường hoặc tôi đang ở nhà.”
 q

10

36

Các quy tắc suy luận

Các chứng minh hình thức



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận
trong logic mệnh đề

11

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận
trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng
minh (Proof
Techniques) bạn nên
tránh

- Cho trước các tiền đề (premises) $p_1, p_2, \dots, p_n$. Một **chứng minh hình thức (formal proof)** của một **kết luận C** là một dãy **các bước (steps)**, trong đó **mỗi bước áp dụng một quy tắc suy luận nào đó cho các tiền đề hoặc các phát biểu đã được chứng minh trước đó** để suy luận ra một phát biểu mới đúng (kết luận)
- Một chứng minh cho thấy rằng **nếu** các tiền đề là đúng, **thì** kết luận là đúng

Ví dụ 25

- Giả sử chúng ta có các tiền đề sau:
 - Trời không nắng và thời tiết lạnh
 - Chúng ta sẽ đi bơi chỉ khi trời nắng
 - Nếu chúng ta không đi bơi, thì chúng ta sẽ đi chèo xuồng
 - Nếu chúng ta đi chèo xuồng, chúng ta sẽ về đến nhà lúc hoàng hôn
- Với các tiền đề đã cho, chứng minh kết luận “Chúng ta sẽ về đến nhà lúc hoàng hôn” bằng cách sử dụng các quy tắc suy luận

Các quy tắc suy luận

Các chứng minh hình thức



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

12

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

■ Bước 1: Xác định các mệnh đề Chúng ta sẽ dùng các ký hiệu sau:

- *sunny* = “Trời nắng”; *cold* = “Thời tiết lạnh”; *swim* = “Chúng ta sẽ đi bơi”; *canoe* = “Chúng ta sẽ đi chèo xuồng”; *sunset* = “Chúng ta sẽ về đến nhà lúc hoàng hôn”

■ Bước 2: Xác định lập luận (Xây dựng dạng cho lập luận)

$p_1 \quad \neg \textit{sunny} \wedge \textit{cold}$ Trời không nắng và thời tiết lạnh

$p_2 \quad \textit{swim} \rightarrow \textit{sunny}$ Chúng ta sẽ đi bơi chỉ khi trời nắng

$p_3 \quad \neg \textit{swim} \rightarrow \textit{canoe}$ Nếu chúng ta không đi bơi, thì chúng ta sẽ đi chèo xuồng

$p_4 \quad \textit{canoe} \rightarrow \textit{sunset}$ Nếu chúng ta đi chèo xuồng, chúng ta sẽ về đến nhà lúc hoàng hôn

$\therefore \quad \textit{sunset}$ Chúng ta sẽ về đến nhà lúc hoàng hôn

Các quy tắc suy luận

Các chứng minh hình thức



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

13 Các chứng minh hình thức

Một số nguyên biến phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

■ Bước 3: Xây dựng chứng minh hoàn chỉnh dựa trên các quy tắc suy luận

Bước	Chứng minh bởi
1. $\neg \text{sunny} \wedge \text{cold}$	Tiền đề p_1
2. $\neg \text{sunny}$	Quy tắc Rút gọn
3. $\text{swim} \rightarrow \text{sunny}$	Tiền đề p_2
4. $\neg \text{swim}$	Modus Tollens cho 2 và 3
5. $\neg \text{swim} \rightarrow \text{canoe}$	Tiền đề p_3
6. canoe	Modus Ponens cho 4 và 5
7. $\text{canoe} \rightarrow \text{sunset}$	Tiền đề p_4
8. sunset	Modus Ponens cho 6 và 7

p_1	$\neg \text{sunny} \wedge \text{cold}$
p_2	$\text{swim} \rightarrow \text{sunny}$
p_3	$\neg \text{swim} \rightarrow \text{canoe}$
p_4	$\text{canoe} \rightarrow \text{sunset}$
$\therefore$	sunset

Các quy tắc suy luận

Một số nguy biến phổ biến



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguy biến phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

- Một *nguy biến (fallacy)* là một quy tắc suy luận hoặc một phương pháp chứng minh không hợp lý về mặt logic
 - Một nguy biến có thể dẫn tới một kết luận sai
- *Nguy biến khẳng định hậu kiện (Fallacy of affirming the conclusion)*
 - $p \rightarrow q$ là đúng, và q là đúng. Do đó, p là đúng (Sai. Bởi vì $\mathbf{F} \rightarrow \mathbf{T}$ đúng)

Ví dụ 26

- Nếu David Cameron là Tổng thống Hoa Kỳ, thì ông ta ít nhất là bốn mươi tuổi ($p \rightarrow q$)
- David Cameron ít nhất là bốn mươi tuổi (q)
- Do đó, David Cameron là Tổng thống Hoa Kỳ (p)
- *Nguy biến phủ định giả thiết (Fallacy of denying the hypothesis)*
 - $p \rightarrow q$ là đúng, và p sai. Do đó q là sai (Sai. Bởi vì $\mathbf{F} \rightarrow \mathbf{T}$ đúng)

Ví dụ 27

- Nếu trời mưa, thì đường lầy lội ($p \rightarrow q$)
- Trời không mưa ($\neg p$)
- Do đó đường không lầy lội ($\neg q$)

14

36

Các quy tắc suy luận

Một số quy tắc suy luận trong logic vị từ



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

■ Quy tắc *Khởi tạo phổ quát (Universal instantiation)*

$$\therefore \frac{\forall x P(x)}{P(c)}$$

Với *bất kỳ một phần tử c cụ thể* trong miền xác định

■ Quy tắc *Tổng quát hóa phổ quát (Universal generalization)*

$$\therefore \frac{P(c)}{\forall x P(x)}$$

Với *bất kỳ một phần tử c nào đó* trong miền xác định

■ Quy tắc *Khởi tạo hiện sinh (Existential instantiation)*

$$\therefore \frac{\exists x P(x)}{P(c)}$$

Với *phần tử c nào đó* trong miền xác định sao cho *P(c) đúng*

■ Quy tắc *Tổng quát hóa hiện sinh (Existential generalization)*

$$\therefore \frac{P(c)}{\exists x P(x)}$$

Với *phần tử c nào đó* trong miền xác định

15

36

Các quy tắc suy luận

Một số quy tắc suy luận trong logic vị từ



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên biện phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Ví dụ 28

- Ta chứng minh lập luận sau: “Mỗi người đều sẽ chết. Socrates là người. Do đó, Socrates sẽ chết”

■ Xác định các vị từ

- $M(x) = “x \text{ là người}”$
- $D(x) = “x \text{ sẽ chết}”$
- $S = “\text{Socrates}”$ – một phần tử trong vũ trụ

■ Xác định lập luận

p_1	$\forall x (M(x) \rightarrow D(x))$	Mỗi người đều sẽ chết
p_2	$M(S)$	Socrates là người
$\therefore$	$D(S)$	Socrates sẽ chết

■ Xây dựng chứng minh

Bước

Chứng minh bởi

- | | |
|--|---------------------------------|
| 1. $\forall x (M(x) \rightarrow D(x))$ | Tiền đề p_1 |
| 2. $M(S) \rightarrow D(S)$ | Quy tắc Khởi tạo phổ quát cho 1 |
| 3. $M(S)$ | Tiền đề p_2 |
| 4. $D(S)$ | Modus Ponens cho 2 và 3 |

16

36

Các quy tắc suy luận

Một số quy tắc suy luận trong logic vị từ



Ví dụ 29

- Lý luận sau đây là đúng hay sai: “Ít nhất một trong số các sinh viên trong lớp rất thông minh. John là một sinh viên trong lớp. Do đó, John rất thông minh”?

■ Xác định các vị từ

- Giả sử *miền xác định* là *tập tất cả mọi người*
- $S(x)$ = “ x là sinh viên trong lớp”
- $I(x)$ = “ x rất thông minh”
- J = “John” – một thành viên trong tập tất cả mọi người

■ Xác định lập luận

$$\begin{array}{ll} p_1 & \exists x (S(x) \wedge I(x)) \quad \text{Ít nhất một trong số các sinh viên trong} \\ & \text{lớp rất thông minh} \\ p_2 & S(J) \quad \text{John là một sinh viên trong lớp} \\ \hline \therefore & I(J) \quad \text{John rất thông minh} \end{array}$$

■ Lập luận có hợp lý không? KHÔNG

- **Phản ví dụ:** Xét trường hợp *có chính xác một sinh viên A trong lớp rất thông minh và A không phải là John*, nghĩa là, $S(A) \wedge I(A)$ đúng, $S(B) \wedge I(B)$ sai với mọi $B \neq A$, và $A \neq J$
- Áp dụng Quy tắc Tổng quát hóa hiện sinh cho $S(A) \wedge I(A)$, *tiền đề* $p_1 = \exists x (S(x) \wedge I(x))$ *đúng*. *Tiền đề* $p_2 = S(J)$ *luôn đúng*
- Tuy nhiên, do $S(B) \wedge I(B)$ sai với mọi $B \neq A$ và $A \neq J$, Quy tắc Khử tạo phổ quát cho ta *kết luận* $I(J)$ *là sai*

Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

17

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Các quy tắc suy luận

Một số quy tắc suy luận trong logic vị từ



Bài tập 32

Lập luận sau là đúng hay sai: “Mọi giảng viên đều ra đề bài kiểm tra khó. Tôi là một giảng viên. Do đó, tôi ra đề bài kiểm tra khó”

Bài tập 33

Với mỗi lập luận sau, hãy giải thích quy tắc suy luận nào được sử dụng trong mỗi bước

- (a) Doug là một sinh viên trong lớp biết cách sử dụng ngôn ngữ lập trình Java. Mỗi người biết sử dụng ngôn ngữ lập trình Java đều có thể tìm được một công việc trả lương cao. Do đó, một sinh viên nào đó trong lớp có thể tìm được một công việc trả lương cao
- (b) Ai đó trong lớp thích xem cá voi. Mỗi người thích xem cá voi đều quan tâm đến vấn đề ô nhiễm đại dương. Do đó, tồn tại một sinh viên trong lớp quan tâm đến vấn đề ô nhiễm đại dương
- (c) Mỗi sinh viên trong lớp có một máy tính cá nhân. Mỗi người có máy tính cá nhân có thể sử dụng một trình soạn thảo văn bản. Do đó, Zeke, một sinh viên trong lớp, có thể sử dụng một trình soạn thảo văn bản

Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

18

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Các quy tắc suy luận

Một số quy tắc suy luận trong logic vị từ



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

19

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Bài tập 34

Lập luận sau để chứng minh nếu $\exists x P(x) \wedge \exists x Q(x)$ đúng thì $\exists x (P(x) \wedge Q(x))$ cũng đúng có hợp lý hay không?

Bước	Chứng minh bởi
1. $\exists x P(x) \wedge \exists x Q(x)$	Tiền đề
2. $\exists x P(x)$	Quy tắc Rút gọn cho 1
3. $P(c)$	Quy tắc khởi tạo hiện sinh cho 2
4. $\exists x Q(x)$	Quy tắc Rút gọn cho 1
5. $Q(c)$	Quy tắc khởi tạo hiện sinh cho 4
6. $P(c) \wedge Q(c)$	Quy tắc hội cho 3 và 5
7. $\exists x P(x) \wedge Q(x)$	Quy tắc Tổng quát hóa hiện sinh

36

Một số lỗi thường gặp



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

20

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Chú ý

Tham khảo từ tài liệu “Common Mistakes in Discrete Mathematics” (https://highered.mheducation.com/sites/dl/free/125967651x/1106131/Common_Mistakes_in_Discrete_Math.pdf)

(a) Dịch sai các phát biểu từ ngôn ngữ thông thường sang dạng ký hiệu

- Ví dụ, có những khó khăn với việc sử dụng từ “hoặc”; hãy chắc chắn phân biệt giữa “ A hoặc B ” ($\equiv A \vee B$) và “hoặc A hoặc B ” ($\equiv A \oplus B$)
- Một phát biểu điều kiện hoàn toàn khác với một phép “hội” ($\wedge$), nhưng một số người không phân biệt được chúng; việc nói rằng “ B sẽ xảy ra nếu A xảy ra” hoàn toàn khác với việc nói rằng “ A và/hoặc B sẽ xảy ra”

36

Một số lỗi thường gặp (tiếp)



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

(b) Phủ định sai các mệnh đề phức hợp do không sử dụng đúng luật De Morgan

- Chẳng hạn, cho rằng $\neg(p \vee q)$ tương đương logic với $\neg p \vee \neg q$, hoặc $\neg(p \wedge q)$ tương đương logic với $\neg p \wedge \neg q$.
- Ví dụ, “không phải là John trên 18 tuổi **hoặc** sống xa nhà” nghĩa là John không trên 18 tuổi **và** (chứ **không phải hoặc**) John không sống xa nhà
- Phát biểu đúng là $\neg(p \vee q)$ tương đương logic với $\neg p \wedge \neg q$, và $\neg(p \wedge q)$ tương đương logic với $\neg p \vee \neg q$
- Lỗi này là một trường hợp tổng quát của việc giả định rằng mỗi toán tử phân phối qua mọi toán tử khác, trong trường hợp này là cho rằng phủ định phân phối qua phép hội (hoặc phép tuyển)

21 Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Một số lỗi thường gặp (tiếp)



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giải thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

22 Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

(c) Viết sai dạng ký hiệu của mệnh đề tồn tại

- Viết sai dạng ký hiệu của mệnh đề tồn tại dưới dạng $\exists x(A(x) \rightarrow B(x))$ thay vì $\exists x(A(x) \wedge B(x))$
- Ví dụ, dạng ký hiệu của “Tồn tại một số chẵn là số nguyên tố” là $\exists x(E(x) \wedge P(x))$, không phải $\exists x(E(x) \rightarrow P(x))$, trong đó $E(x)$ có nghĩa là “ x là số chẵn” và $P(x)$ có nghĩa là “ x là số nguyên tố”
- Thông thường, các lượng từ tồn tại thường được theo sau bởi các phép hội

(d) Viết sai dạng ký hiệu của mệnh đề phổ quát

- Viết sai dạng ký hiệu của mệnh đề phổ quát dưới dạng $\forall x(A(x) \wedge B(x))$ thay vì $\forall x(A(x) \rightarrow B(x))$
- Ví dụ, dạng ký hiệu của “Mọi số lẻ là số nguyên tố” là $\forall x(O(x) \rightarrow P(x))$, không phải $\forall x(O(x) \wedge P(x))$, trong đó $O(x)$ có nghĩa là “ x là số lẻ” và $P(x)$ có nghĩa là “ x là số nguyên tố”
- Thông thường, các lượng từ phổ quát thường được theo sau bởi các mệnh đề điều kiện

Một số lỗi thường gặp (tiếp)



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

23

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

(e) Đặt sai vị trí các vị từ bên trong vị từ khác

- Đặt sai vị trí các vị từ bên trong vị từ khác, ví dụ như $P(Q(x))$
- Ví dụ, nếu $P(x)$ là “ x là số nguyên tố,” và $Q(x)$ là “ x là số lẻ,” thì việc viết $P(Q(x))$ để cố gắng biểu diễn một phát biểu như “ x là số nguyên tố lẻ” hoặc viết $\forall x P(Q(x))$ để nói “mọi số lẻ đều là số nguyên tố” là vô nghĩa
- Ký hiệu $P(Q(x))$ thực sự có nghĩa là *phát biểu “ x là số lẻ” là một số nguyên tố* (hay *phát biểu $Q(x)$ là một số nguyên tố*), và rõ ràng một phát biểu không phải là bất kỳ loại số nào

(f) Không thay đổi lượng từ khi phủ định một mệnh đề có lượng từ

- Ví dụ, phủ định của phát biểu “*một số* con mèo *thích* gan lợn” không phải là “*một số* con mèo *không thích* gan lợn”; mà là “*không* có con mèo nào *thích* gan lợn”, hoặc “*tất cả* các con mèo đều *không thích* gan lợn”

Một số lỗi thường gặp (tiếp)



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

24

Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

(g) Lạm dụng thuật ngữ “theo định nghĩa” khi biện minh cho các phát biểu trong chứng minh

- Ví dụ, Franklin Roosevelt không phải là Tổng thống Hoa Kỳ vào thời điểm nước này tham gia Thế chiến thứ II vào tháng 12 năm 1941. Tuy nhiên, “theo định nghĩa”, ông là Tổng thống vì ông đã được bầu vào vị trí này vào đầu năm 1941 và chưa mất hay rời khỏi chức vụ

(h) Không kiểm tra cẩn thận các định nghĩa trong chứng minh

- Ví dụ, nếu đang cố gắng chứng minh điều gì đó về các số nguyên lẻ, thì việc sử dụng chính xác ý nghĩa của khái niệm đó (rằng một số nguyên lẻ là một số có thể viết dưới dạng $2k + 1$ với k là một số nguyên) tại một hoặc nhiều chỗ trong chứng minh là rất quan trọng

(i) Bắt đầu chứng minh bằng cách giả định điều cần chứng minh

Một số lỗi thường gặp (tiếp)



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

25 Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

- Một trường hợp phổ biến của lỗi này là *cố gắng chứng minh các đẳng thức bằng cách bắt đầu với chính đẳng thức đó và sử dụng các biến đổi đại số để đạt được $A = A$* ; cách này là *không hợp lệ*. Tương tự, nếu chúng ta đang cố gắng chứng minh một đẳng thức tập hợp trong phần “Các cấu trúc cơ bản”, chẳng hạn như $A \subseteq A \cup B$, sẽ là *không hợp lệ* khi *bắt đầu với chính phát biểu $A \subseteq A \cup B$*

(j) Giả định sai rằng một vài ví dụ của một mệnh đề phổ quát có nghĩa là mệnh đề đó đúng

- Giả định không hợp lệ rằng một vài (hoặc thậm chí số lượng lớn) ví dụ của một mệnh đề phổ quát có nghĩa là mệnh đề đó đúng
- Có một ví dụ từ lý thuyết số về một mệnh đề thú vị về số nguyên dương n mà đúng với mọi $n \leq 4\,000\,000$ ngoại trừ trường hợp $n = 1969$
- Một chứng minh của mệnh đề phổ quát $\forall x P(x)$ bao gồm việc chỉ ra rằng thuộc tính $P(x)$ luôn đúng bất kể x được chọn như thế nào từ miền xác định (vũ trụ các đối tượng)

Một số lỗi thường gặp (tiếp)



Lôgic và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong lôgic mệnh đề

Các chứng minh hình thức

Một số nguyên biện phổ biến

Một số quy tắc suy luận trong lôgic vị từ

26 Một số lỗi thường gặp

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

(k) Giả định sai rằng đối tượng tùy ý có thuộc tính cụ thể khi chỉ biết tồn tại một đối tượng với thuộc tính đó

- Ví dụ, giả sử chúng ta đang cố gắng chứng minh khẳng định rằng x^2 luôn dư 1 khi chia cho 8
- Sẽ là không hợp lệ khi bắt đầu chứng minh bằng cách giả sử rằng $x = 2n + 1$ với một số nguyên n nào đó
- Mặc dù một số số nguyên thỏa mãn tính chất này là số lẻ, nhưng không phải tất cả số nguyên đều vậy, vì vậy chúng ta sẽ chỉ chứng minh khẳng định đúng trong một số trường hợp, chứ không phải luôn luôn đúng

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

27

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Chú ý

Đây là bản dịch tài liệu *Proof Techniques* của Dana Angluin (SIGACT News, Winter-Spring 1983, Volume 15 #1). Bản dịch này tuân theo một bản sao của tài liệu ở <https://mfleck.cs.illinois.edu/proof.html>.

Mọi sai sót trong bản dịch này hoàn toàn là do hạn chế về kiến thức của người dịch. Mọi góp ý xin gửi về hoanganhduc@hus.edu.vn.

Chứng minh bằng ví dụ Các tác giả đưa ra chứng minh cho $n = 2$ và đề nghị rằng nó có chứa phần lớn các ý tưởng của chứng minh cho trường hợp tổng quát.

Chứng minh bằng hăm dọa “Tầm thường” hoặc “hiển nhiên”.

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh (tiếp)



Chứng minh bằng cách liệt kê tất cả mọi thứ Một hoặc hai số của một tạp chí chỉ dành riêng cho chứng minh của bạn là một điều có ích.

Chứng minh bằng cách bớt xén “Độc giả có thể dễ dàng đưa ra các chi tiết”, “253 trường hợp còn lại được tiến hành tương tự”.

Chứng minh bằng cách giấu giếm Một chuỗi dài các phát biểu liên quan đúng và/hoặc vô nghĩa về mặt cú pháp.

Chứng minh bằng cách trích dẫn đầy ước muốn Tác giả trích dẫn phủ định, đảo, hoặc tổng quát của một định lý đã biết để hỗ trợ cho khẳng định của mình.

Chứng minh bằng hỗ trợ tài chính Làm sao mà ba tổ chức chính phủ khác nhau có thể sai được? Hoặc theo một góc nhìn đối lập: làm sao mà bất cứ thứ gì hỗ trợ tài chính bởi những tổ chức tầm thường này có thể đúng được?

Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

28

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh (tiếp)



Chứng minh bằng dân chủ Rất nhiều người tin rằng điều này đúng: làm sao mà tất cả bọn họ đều sai được?

Chứng minh bằng kinh tế thị trường Lý thuyết của tôi là lý thuyết duy nhất trên thị trường sẽ xử lý các dữ liệu.

Chứng minh bằng sự hiểu biết sâu sắc “Tôi thấy Ruzena ở trong thang máy và cô ấy nói rằng điều đó đã được thử nghiệm ở những năm 1970 và không dùng được”.

Chứng minh bằng vũ trụ Phủ định của mệnh đề này là không tưởng hoặc vô nghĩa. Phổ biến cho các chứng minh về sự tồn tại của Chúa và cho các chứng minh rằng máy tính không thể suy nghĩ.

Chứng minh bằng liên hệ cá nhân “Bài toán Eight-dimensional colored cycle stripping là NP-đầy đủ [Karp, liên hệ cá nhân]”.

Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

29

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh (tiếp)



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

30

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Chứng minh bằng cách liên hệ đến các bài nói chuyện “Ở một buổi hội thảo đặc biệt của NSA về lĩnh vực thị giác máy tính, Binford đã chứng minh rằng SHGC không thể nhận biết được trong thời gian đa thức”.

Chứng minh bằng cách đưa về sai bài toán “Để thấy rằng bài toán infinite-dimensional coloured cycle stripping có thể giải được, ta đưa nó về bài toán halting”.

Chứng minh bằng cách trích dẫn các nguồn không truy cập được
Tác giả trích dẫn một hệ quả đơn giản của một định lý được tìm ra trong một bản ghi nhớ lưu hành nội bộ của Hiệp hội Triết học Slovenia năm 1883. Phương pháp này thậm chí còn hiệu quả hơn nếu tài liệu này chưa bao giờ được dịch từ bản gốc tiếng Iceland.

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh (tiếp)



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

31

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Chứng minh bằng cách trích dẫn các nguồn không tồn tại

Không có điều gì thậm chí hơi giống với định lý đã được trích dẫn xuất hiện ở trong tài liệu được đề cập. Tốt hơn là kết hợp với phương pháp chứng minh bằng cách trích dẫn các nguồn không truy cập được.

Chứng minh bằng cách trích dẫn một tài liệu sẽ xuất bản

Thông thường tác giả sẽ trích dẫn một bài báo sắp xuất bản của chính mình, và tài liệu này thường không còn sắp xuất bản như lúc đầu.

Chứng minh bằng tính quan trọng Một lượng lớn các hệ quả hữu ích đều suy ra từ mệnh đề trong câu hỏi.

Chứng minh bằng việc tích lũy bằng chứng Việc tìm kiếm lâu dài và siêng năng không cho ta bất kỳ một phản ví dụ nào.

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh (tiếp)



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

Chứng minh bằng các tài liệu tham khảo lẫn nhau Trong tài liệu A, Định lý 5 được cho là suy ra từ Định lý 3 của tài liệu B, và định lý này được suy ra từ Hệ quả 6.2 trong tài liệu C, và hệ quả này là một hệ quả dễ dàng suy ra được từ Định lý 5 của tài liệu A.

Chứng minh bằng siêu chứng minh Một phương pháp được đưa ra để xây dựng chứng minh. Tính đúng đắn của phương pháp này được chứng minh bằng bất kể một kỹ thuật nào trong số các kỹ thuật này. Sự hiểu biết sâu sắc về ngữ nghĩa ngôn ngữ lập trình sẽ giúp ích ở đây.

Chứng minh bằng hình vẽ Một hình thức thuyết phục hơn của Chứng minh bằng ví dụ. Kết hợp tốt với Chứng minh bằng cách bớt xén.

32 Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh (tiếp)



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

36 Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Chứng minh bằng đồ họa hào nhoáng Còn được gọi là phương pháp Jabberwocky. Chỉ có một kết quả thực sự mạnh mẽ mới có thể làm nền tảng cho một màn trình diễn âm thanh và ánh sáng tuyệt vời như vậy. “Sản phẩm dành cho những người không có bài thuyết trình.”

Chứng minh bằng các biểu đồ dễ gây nhầm lẫn hoặc không giải thích được Hầu như bất kỳ đường cong nào cũng có thể được tạo ra để trông giống như kết quả mong muốn bằng cách chuyển đổi phù hợp các biến và thao tác với các tỷ lệ trục. Thường gặp trong công việc thí nghiệm.

Chứng minh bằng việc khẳng định một cách kịch liệt Tốt nhất là khi có một loại quan hệ quyền hạn nào đó với khán giả, và do đó phương pháp này đặc biệt hữu ích trong một lớp học.

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh (tiếp)



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

34

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Chứng minh bằng cách lặp lại Cũng được biết đến như là chứng minh của Bellman: “Điều gì tôi nói ba lần là đúng.”

Chứng minh bằng cách kêu gọi trực giác Các hình vẽ theo dạng đám mây thường giúp ích ở đây.

Chứng minh bằng cách vẩy tay một cách mạnh mẽ Hoạt động tốt trong môi trường lớp học, xêmina hoặc hội thảo.

Chứng minh bằng cách thay đổi ngữ nghĩa Một số định nghĩa cơ bản nhưng bất tiện được thay đổi để phù hợp với phát biểu của kết quả.

Chứng minh bằng ký hiệu rườm rà Tốt nhất là thực hiện với việc sử dụng ít nhất bốn bảng chữ cái, các ký tự đặc biệt, và phiên bản mới nhất của LaTeX.

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh (tiếp)



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

35

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Chứng minh bằng sự trừu tượng vô nghĩa Một phiên bản của Chứng minh bằng hăm dọa. Tác giả sử dụng các thuật ngữ hoặc định lý từ toán học cao cấp trong rất ấn tượng nhưng chỉ liên quan trực tiếp đến vấn đề hiện tại. Một vài tích phân ở đây, một vài dãy số chính xác ở kia, và ai sẽ biết liệu bạn có thực sự có chứng minh hay không?

Phản chứng bằng cách tìm ra một quả táo xấu Một quả táo xấu làm hỏng cả chùm¹. Trong số nhiều người ủng hộ lý thuyết này, chúng tôi đã tìm thấy một người rõ ràng là điên rồ; vì vậy chúng ta có thể làm mất uy tín của toàn bộ lý thuyết. (Thường sử dụng trong ngữ cảnh chính trị.)

36

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh (tiếp)



Lógica và Chứng minh

Hoàng Anh Đức

Các quy tắc suy luận

Giới thiệu

Một số quy tắc suy luận trong logic mệnh đề

Các chứng minh hình thức

Một số nguyên tắc phổ biến

Một số quy tắc suy luận trong logic vị từ

Một số lỗi thường gặp

36

Các kỹ thuật chứng minh (Proof Techniques) bạn nên tránh

Chứng minh bằng con đường dốc trơn trượt Nếu chúng tôi chấp nhận [đề xuất ban đầu], chúng tôi sẽ phải chấp nhận [đề xuất được sửa đổi một chút] và cuối cùng điều này sẽ dẫn đến [đề xuất hoàn toàn khác biệt và rõ ràng là có thể bị phản đối].

Chứng minh bằng “không phát minh ở đây” Chúng tôi có kinh nghiệm làm việc với thiết bị này trong nhiều năm ở MIT và chúng tôi chưa bao giờ nhận ra hiệu quả này.

¹ Con sâu làm rầu nồi canh

1. Logic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Đệ quy
4. Thuật toán I: Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp
5. Thuật toán II: Thuật toán tham lam, thuật toán đệ quy
6. Lý thuyết số cơ bản
7. Các phương pháp đếm
8. Lý thuyết đồ thị I: Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông
9. Lý thuyết đồ thị II: Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị
10. Lý thuyết đồ thị III: Cây
11. Đại số Boole

VNU-HUS MAT3500: Toán rời rạc

Các cấu trúc cơ bản Tập hợp, Hàm, Dãy, Tổng/Tích

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn



Nội dung



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản
Các phép toán trên tập hợp
Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ
Định nghĩa hàm và một số khái niệm
Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm
Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm
Một số công thức tổng hữu ích
Ký hiệu tích

Tập hợp

Một số khái niệm và tính chất cơ bản
Các phép toán trên tập hợp
Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ
Định nghĩa hàm và một số khái niệm
Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm
Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm
Một số công thức tổng hữu ích
Ký hiệu tích

Tập hợp

Khái niệm và cách mô tả tập hợp



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

2

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Đãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- Một **tập hợp (set)** là một tổng thể không sắp thứ tự các đối tượng phân biệt (gọi là các **phần tử (element)** hoặc **thành viên (member)** của tập hợp)
 - $x \in S$: x là phần tử của S
 - $x \notin S$: x không là phần tử của S
- Ta thường sử dụng các chữ in hoa $S, T, U, \dots$ để ký hiệu tập hợp
- Có thể mô tả một tập hợp bằng cách **liệt kê tất cả các phần tử** của tập đó giữa hai dấu ngoặc nhọn “{” và “}”. Trong nhiều trường hợp, có thể **liệt kê thông qua “quy luật đơn giản”**
 - Tập các nguyên âm trong bảng chữ cái tiếng Anh
 $V = \{a, e, i, o, u\}$
 - Tập các số tự nhiên $\mathbb{N} = \{0, 1, 2, 3, \dots\}$
- Có thể mô tả một tập hợp thông qua **quy tắc nhận biết**
 - Với vị từ $P(x)$ bất kỳ trên miền xác định nào đó, $\{x \mid P(x)\}$ là tập hợp tất cả x sao cho $P(x)$ đúng (có thể dùng “:” thay vì “|”)
 - Tập các số tự nhiên chẵn $E = \{x \mid x = 2k \text{ với } k \in \mathbb{N}\}$

Tập hợp

Khái niệm và cách mô tả tập hợp



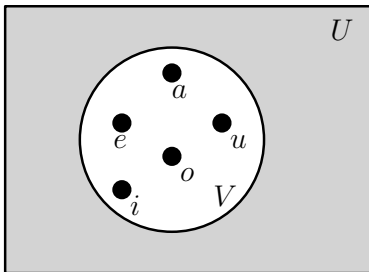
Có thể mô tả một (hoặc nhiều) tập hợp thông qua **giản đồ Venn** (*Venn diagram*)

- **Tập vũ trụ (universal set) U** gồm tất cả các đối tượng đang xét
- Tập hợp cần mô tả
- Phần tử của tập hợp

Hình chữ nhật

Hình tròn hoặc các hình khác

Điểm



Hình: Mô tả tập các nguyên âm trong bảng chữ cái tiếng Anh $V = \{a, e, i, o, u\}$ bằng giản đồ Venn

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

3 Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Tập hợp

Tập hợp rỗng



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

4

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- **Tập hợp rỗng (empty set)**, ký hiệu $\emptyset$, là tập hợp duy nhất không chứa bất kỳ phần tử nào
- $\emptyset = \{\}$ hoặc $\emptyset = \{x \mid \mathbf{F}\}$ với $\mathbf{F}$ là một mệnh đề luôn luôn sai (mâu thuẫn)
- Bất kể miền xác định là gì, **mệnh đề $\neg \exists x (x \in \emptyset)$ luôn đúng**
- $\emptyset \neq \{\emptyset\}$
 - Tập $\{\emptyset\}$ không rỗng, vì nó chứa một phần tử—tập hợp rỗng

Tập hợp

Tập hợp con và tập hợp bằng nhau



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

5

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

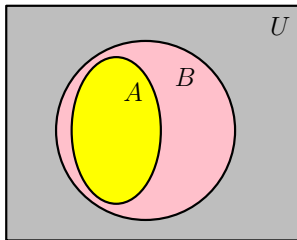
Cho hai tập hợp A và B . A là **tập con (subset)** của B , ký hiệu $A \subseteq B$ hoặc $B \supseteq A$, khi và chỉ khi mỗi phần tử của tập A cũng là một phần tử của B

- $(A \subseteq B) \equiv \forall x (x \in A \rightarrow x \in B)$
- $(A \not\subseteq B) \equiv \neg(A \subseteq B)$ (A **không** là tập con của B)
- $(A \subset B) \equiv (A \subseteq B) \wedge (B \not\subseteq A)$ (A là **tập con thực sự (proper subset)** của B)

Bài tập 1

Cho $A = \{1, 2, 3\}$ và $B = \{1, 3, 5, 7\}$. Hãy liệt kê tất cả các tập hợp

- (a) là tập con của A
- (b) là tập con thực sự của A
- (c) vừa là tập con của A vừa là tập con của B
- (d) là tập con của A nhưng không là tập con của B



Hình: $A \subset B$

Tập hợp

Tập hợp con và tập hợp bằng nhau



Các cấu trúc cơ bản

Hoàng Anh Đức

Ví dụ 1

Ta chứng minh *với mọi tập A , ta có $\emptyset \subseteq A$*

- Nghĩa là, ta cần chứng minh $\forall A \forall x ((x \in \emptyset) \rightarrow (x \in A))$
- Cụ thể, ta cần chỉ ra rằng với tập hợp A_0 và phần tử x_0 thuộc các miền xác định tương ứng, mệnh đề $(x_0 \in \emptyset) \rightarrow (x_0 \in A_0)$ luôn đúng
- Thật vậy, theo định nghĩa của tập hợp rỗng, $(x_0 \in \emptyset) = F$.
Do đó, $(x_0 \in \emptyset) \rightarrow (x_0 \in A_0) = T$

Bài tập 2

Phát biểu sau đúng hay sai?

“Với mọi tập hợp A , nếu tồn tại tập hợp B sao cho $B \subseteq A$, thì $A \neq \emptyset$ ”

Bài tập 3

Chứng minh rằng với mọi tập hợp A, B, C , nếu $A \subseteq B$ và $B \subseteq C$ thì $A \subseteq C$

Bài tập 4

Chứng minh rằng với mọi tập hợp A , ta có $A \subseteq A$

Bài tập 5

Liệu có tồn tại các tập hợp A và B thỏa mãn $A \in B$ và $A \subseteq B$?

6

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Tập hợp

Tập hợp con và tập hợp bằng nhau



Các cấu trúc cơ bản

Hoàng Anh Đức

Cho hai tập hợp A và B . A và B là hai tập **bằng nhau**, ký hiệu $A = B$, khi và chỉ khi $A \subseteq B$ và $B \subseteq A$

- $(A = B) \equiv (A \subseteq B) \wedge (B \subseteq A) \equiv \forall x (x \in A \leftrightarrow x \in B)$
- Tất cả các phần tử trong một tập đều **phân biệt (distinct)**; liệt kê một phần tử nhiều lần là vô nghĩa
 - Nếu $a = b$ thì $\{a, b, c\} = \{a, c\} = \{b, c\} = \{a, a, b, c, a, c, c\}$
 - Ta nói rằng tập trên có (nhiều nhất) 2 phần tử
- Các phần tử của một tập hợp **không sắp thứ tự (unordered)**
 - Bất kể a, b, c là gì, $\{a, b, c\} = \{a, c, b\} = \{b, a, c\} = \{b, c, a\} = \{c, a, b\} = \{c, b, a\}$

7

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Tập hợp

Lực lượng của một tập hợp



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

8

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- **Lực lượng (cardinality)** của một tập A , ký hiệu $|A|$, là số phần tử khác biệt mà A có

- $|\emptyset| = 0$; $|\{1, 2, 3\}| = 3$; $|\{\{1, 2, 3\}, \{4, 5\}\}| = 2$

- Nếu $|A| \in \mathbb{N}$, thì ta gọi A là **tập hữu hạn (finite set)**. Ngược lại, A là một **tập vô hạn (infinite set)**

- Một số tập vô hạn quan trọng

- $\mathbb{N} = \{0, 1, 2, \dots\}$ **Tập số tự nhiên (natural numbers)**

- $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ **Tập số nguyên (integers)**

- $\mathbb{Z}^+ = \{1, 2, \dots\}$ **Tập số nguyên dương (positive integers)**

- $\mathbb{Q} = \{p/q \mid p, q \in \mathbb{Z}, \text{ và } q \neq 0\}$ **Tập số hữu tỷ (rational numbers)**

- $\mathbb{R}$ **Tập số thực (real numbers)**

- $\mathbb{R}^+$ **Tập số thực dương (positive real numbers)**

- $\mathbb{C}$ **Tập số phức (complex numbers)**

- $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$

Bài tập 6

Tìm các tập hợp A, B thỏa mãn các điều kiện

(a) $A = \{3, |B|\}$ và (b) $B = \{1, |A|, |B|\}$

- **Tập lũy thừa (power set)** của một tập A , ký hiệu $\mathcal{P}(A)$, là tập hợp gồm tất cả các tập con của A

- $\mathcal{P}(A) = \{x \mid x \subseteq A\}$
- $\mathcal{P}(\{a, b\}) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$
- $\mathcal{P}(\emptyset) = \{\emptyset\}$
- $\mathcal{P}(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}$

- Nếu A là tập hữu hạn, $|\mathcal{P}(A)| = 2^{|A|}$. Do đó ký hiệu 2^A đôi khi cũng được sử dụng để chỉ tập lũy thừa của A

Bài tập 7

Cho $A = \{1, 2, 3\}$. Các mệnh đề sau đúng hay sai?

- (a) $2 \in A$ (c) $2 \in \mathcal{P}(A)$ (e) $\{2\} \in \mathcal{P}(A)$ (g) $\{\{2\}\} \in \mathcal{P}(A)$
(b) $2 \subseteq A$ (d) $2 \subseteq \mathcal{P}(A)$ (f) $\{2\} \subseteq \mathcal{P}(A)$ (h) $\{\{2\}\} \subseteq \mathcal{P}(A)$

Bài tập 8

Chứng minh rằng nếu $A = B$ thì $\mathcal{P}(A) = \mathcal{P}(B)$ với hai tập A, B bất kỳ. Ngược lại, nếu $\mathcal{P}(A) = \mathcal{P}(B)$ thì A có bằng B không?

(Gợi ý: $A = B \equiv (A \subseteq B) \wedge (B \subseteq A) \equiv \forall x (x \in A \leftrightarrow x \in B)$ và nếu $A \subseteq B$ và $B \subseteq C$ thì $A \subseteq C$)

Tập hợp

Tích Đềcắc



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

10

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- Với $n \in \mathbb{N}$, một **bộ sắp thứ tự n phần tử (ordered n -tuples)** $(a_1, a_2, \dots, a_n)$ là một dãy các phần tử có phần tử thứ nhất là a_1 , phần tử thứ hai là $a_2, \dots$, và phần tử thứ n là a_n
 - Một bộ sắp thứ tự 2 phần tử được gọi là một **cặp sắp thứ tự (order pair)**
- Hai bộ $(a_1, \dots, a_n)$ và $(b_1, \dots, b_n)$ là **bằng nhau** nếu với mọi $i \in \{1, \dots, n\}$, $a_i = b_i$
- **Chú ý:** $(1, 2) \neq (2, 1) \neq (2, 1, 1)$ nhưng $\{1, 2\} = \{2, 1\} = \{2, 1, 1\}$
- **Tích Đềcắc (Cartesian product)** của hai tập A, B , ký hiệu $A \times B$, là tập tất cả các cặp sắp thứ tự (a, b) trong đó $a \in A$ và $b \in B$
 - $A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$
 - Chú ý rằng tích Đềcắc **không** có tính chất giao hoán, nghĩa là $\neg \forall A, B (A \times B = B \times A)$
 - Tổng quát hóa
$$A_1 \times \dots \times A_n = \{(a_1, \dots, a_n) \mid a_1 \in A_1 \wedge \dots \wedge a_n \in A_n\}$$

Tập hợp

Tích Đề các



Các cấu trúc cơ bản

Hoàng Anh Đức

Bài tập 9

Cho $A = \{1, \{2, 3\}\}$ và $B = \{4, 5, 6\}$. Tìm các tập hợp $A \times A$, $B \times B$, $A \times B$, và $B \times A$

Bài tập 10

Chứng minh rằng $A \times B = \emptyset$ khi và chỉ khi $A = \emptyset$ hoặc $B = \emptyset$

Bài tập 11

Chứng minh rằng $A \times B = B \times A$ khi và chỉ khi $A = \emptyset$ hoặc $B = \emptyset$ hoặc $A = B$

11

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích



Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

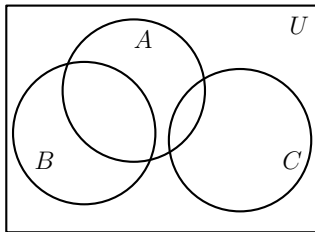
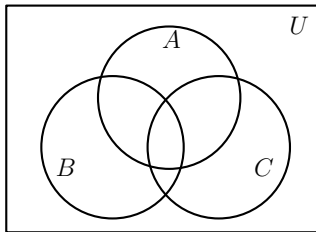
Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

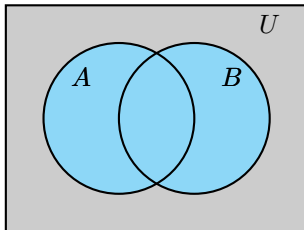
- Biểu đồ Venn có thể được sử dụng để mô tả mối quan hệ giữa hai hoặc nhiều tập hợp
- Trong *biểu đồ Venn với hai tập hợp trở lên*, các đường cong (vòng tròn hoặc các hình khác) được chồng lấp theo mọi cách có thể, *thể hiện tất cả các mối quan hệ có thể giữa các tập.*



12

■ **Hợp (union)** của hai tập hợp A, B , ký hiệu $A \cup B$, là tập chứa tất cả các phần tử hoặc thuộc A , hoặc thuộc B , hoặc thuộc cả hai

- $\forall A, B (A \cup B = \{x \mid x \in A \vee x \in B\})$
- $A \cup B \supseteq A$ và $A \cup B \supseteq B$
- $\{1, 3, 5\} \cup \{2, 3, 4\} = \{1, 2, 3, 4, 5\}$



Hình: Biểu đồ Venn mô tả $A \cup B$

- **Giao (intersection)** của hai tập hợp A, B , ký hiệu $A \cap B$, là tập chứa tất cả các phần tử đồng thời thuộc cả A và B

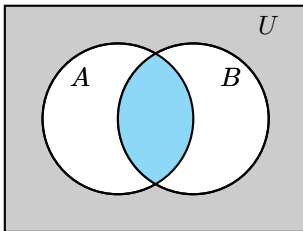
- $\forall A, B (A \cap B = \{x \mid x \in A \wedge x \in B\})$

- $A \cap B \subseteq A$ và $A \cap B \subseteq B$

- $\{1, 3, 5\} \cap \{2, 3, 4\} = \{3\}$

- Hai tập A và B là **rời nhau (disjoint)** nếu $A \cap B = \emptyset$.

- $\{1, 3, 5\} \cap \{2, 4, 6\} = \emptyset$



Hình: Biểu đồ Venn mô tả $A \cap B$

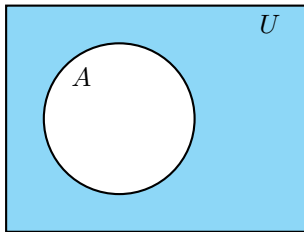
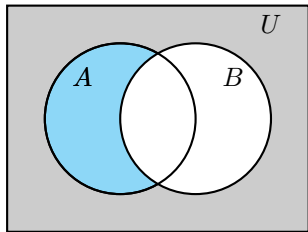
- **Hiệu (difference)** của hai tập hợp A, B , ký hiệu $A - B$ hoặc $A \setminus B$, là tập chứa tất cả các phần tử thuộc A nhưng không thuộc B

- $\forall A, B (A - B = \{x \mid x \in A \wedge x \notin B\})$

- $\{1, 3, 5\} - \{2, 3, 4\} = \{1, 5\}$

- Khi tập vũ trụ U được xác định, **phần bù (complement)** của tập A , ký hiệu $\bar{A}$, là tập $U - A$

- $\forall A (\bar{A} = \{x \mid x \notin A\})$



Hình: Giản đồ Venn mô tả $A - B$

Hình: Giản đồ Venn mô tả $\bar{A}$

Tập hợp

Phép hiệu đối xứng



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

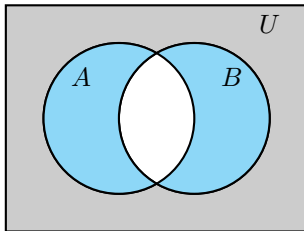
Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- **Hiệu đối xứng (symmetric difference)** của hai tập hợp A, B , ký hiệu $A \Delta B$ hoặc $A \oplus B$, là tập chứa tất cả các phần tử hoặc thuộc A hoặc thuộc B nhưng không thuộc cả A và B

- $\forall A, B (A \Delta B = \{x \mid x \in A \oplus x \in B\})$
- $A \Delta B = (A - B) \cup (B - A)$
- $\{1, 3, 5\} \Delta \{2, 3, 4\} = \{1, 2, 4, 5\}$



Hình: Biểu đồ Venn mô tả $A \Delta B$

16

78

Tập hợp

Các phép toán trên tập hợp



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

17

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Bài tập 12

Tìm các tập A và B , biết rằng $A - B = \{1, 5, 7, 8\}$, $B - A = \{2, 10\}$, và $A \cap B = \{3, 6, 9\}$

Bài tập 13

Cho các tập hợp A, B . Chứng minh

(a) $A \cap B \subseteq A$ và $A \cap B \subseteq B$

(b) $A \subseteq (A \cup B)$

(c) $A - B \subseteq A$

Bảng tính thuộc (membership table) của các phép toán trên tập hợp

A	B	$A \cup B$	$A \cap B$	$A - B$	$\overline{A}$	$A \Delta B$
1	1	1	1	0	0	0
1	0	1	0	1	0	1
0	1	1	0	0	1	1
0	0	0	0	0	1	0

Bài tập 14

Xây dựng bảng tính thuộc của

- (a) $A \cup (B \cup C)$ và $(A \cup B) \cup C$
- (b) $A \cap (B \cup C)$ và $(A \cap B) \cup (A \cap C)$
- (c) $\overline{A \cup B}$ và $\overline{A} \cap \overline{B}$

Tập hợp

Một số khái niệm và tính chất cơ bản

18

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Tập hợp

Các hằng đẳng thức tập hợp



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Tên gọi	Đẳng thức
Luật đồng nhất (Identity laws)	$A \cap U = A$ $A \cup \emptyset = A$
Luật nuốt (Domination laws)	$A \cup U = U$ $A \cap \emptyset = \emptyset$
Luật lũy đẳng (Idempotent laws)	$A \cup A = A$ $A \cap A = A$
Luật bù kép (Double complement laws)	$\overline{\overline{A}} = A$
Luật giao hoán (Commutative laws)	$A \cup B = B \cup A$ $A \cap B = B \cap A$
Luật kết hợp (Associative laws)	$A \cup (B \cap C) = (A \cup B) \cap C$ $A \cap (B \cup C) = (A \cap B) \cup C$
Luật phân phối (Distributive laws)	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Tập hợp

Các hằng đẳng thức tập hợp



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

20

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Tên gọi	Đẳng thức
Luật De Morgan (De Morgan's laws)	$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$
Luật hấp thụ (Absorption laws)	$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$
Luật bù (Complement laws)	$A \cup \overline{A} = U$ $A \cap \overline{A} = \emptyset$

Với hai tập A, B bất kỳ,

Chứng minh $A = B$

- (1) Chứng minh trực tiếp $A \subseteq B$ và $B \subseteq A$
- (2) Chứng minh thông qua định nghĩa tập hợp và các phép biến đổi logic
- (3) Chứng minh bằng bảng tính thuộc

Tập hợp

Các hằng đẳng thức tập hợp



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

21

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Đây

Định nghĩa dây và một số khái niệm

Một số dây đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Ví dụ 2 (Dùng định nghĩa)

Chứng minh $\overline{A \cap B} = \overline{A} \cup \overline{B}$

■ $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$

- Giả sử $x \in \overline{A \cap B}$. Theo định nghĩa, $x \notin A \cap B$. Do đó, mệnh đề $\neg(x \in A \wedge x \in B)$ đúng. Áp dụng luật De Morgan, $\neg(x \in A) \vee \neg(x \in B)$ đúng. Theo định nghĩa, ta có $x \notin A$ hoặc $x \notin B$. Do đó, $x \in \overline{A}$ hoặc $x \in \overline{B}$, suy ra $x \in \overline{A} \cup \overline{B}$

■ $\overline{A \cap B} \supseteq \overline{A} \cup \overline{B}$

- Giả sử $x \in \overline{A} \cup \overline{B}$. Theo định nghĩa, $x \in \overline{A}$ hoặc $x \in \overline{B}$. Do đó, $x \notin A$ hoặc $x \notin B$. Như vậy, mệnh đề $(x \notin A) \vee (x \notin B)$ đúng. Theo định nghĩa, $\neg(x \in A) \vee \neg(x \in B)$ cũng đúng. Áp dụng luật De Morgan, mệnh đề $\neg(x \in A \wedge x \in B)$ đúng. Do đó, $\neg(x \in A \cap B)$ đúng, suy ra $x \in \overline{A \cap B}$

Tập hợp

Các hằng đẳng thức tập hợp



Các cấu trúc cơ bản

Hoàng Anh Đức

Ví dụ 3 (Dùng đẳng thức logic đã biết)

Chứng minh $\overline{A \cap B} = \overline{A} \cup \overline{B}$

$$\begin{aligned}\overline{A \cap B} &= \{x \mid x \notin A \cap B\} \\ &= \{x \mid \neg(x \in A \cap B)\} \\ &= \{x \mid \neg(x \in A \wedge x \in B)\} \\ &= \{x \mid \neg(x \in A) \vee \neg(x \in B)\} \\ &= \{x \mid x \notin A \vee x \notin B\} \\ &= \{x \mid x \in \overline{A} \vee x \in \overline{B}\} \\ &= \{x \mid x \in \overline{A} \cup \overline{B}\} \\ &= \overline{A} \cup \overline{B}\end{aligned}$$

định nghĩa phần bù

định nghĩa $\neg$

định nghĩa $\cap$

luật De Morgan

định nghĩa $\vee$

định nghĩa phần bù

định nghĩa $\cup$

mô tả tập hợp

Tập hợp

Một số khái niệm và tính chất cơ bản

22

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Tập hợp

Các hằng đẳng thức tập hợp



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

23

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Ví dụ 4 (Dùng bảng tính thuộc)

Chứng minh $\overline{A \cap B} = \overline{A} \cup \overline{B}$

A	B	$\overline{A}$	$\overline{B}$	$A \cap B$	$\overline{A \cap B}$	$\overline{A} \cup \overline{B}$
1	1	0	0	1	0	0
1	0	0	1	0	1	1
0	1	1	0	0	1	1
0	0	1	1	0	1	1

Tập hợp

Các hằng đẳng thức tập hợp



Bài tập 15

Chứng minh các hằng đẳng thức tập hợp đã đề cập (sử dụng các phương pháp đã trình bày)

Bài tập 16

Với các tập A, B bất kỳ, chứng minh

$$(a) A \cap B = A - (A - B)$$

$$(d) A \Delta A = \emptyset$$

$$(b) A \cup (B - A) = A \cup B$$

$$(e) A \Delta \emptyset = A$$

$$(c) A \cap (B - A) = \emptyset$$

$$(f) A \Delta B = B \Delta A$$

Bài tập 17

Với các tập A, B, C , có thể kết luận rằng $A = B$ nếu

$$(a) A \cup C = B \cup C?$$

$$(b) A \cap C = B \cap C?$$

$$(c) A \cup C = B \cup C \text{ và } A \cap C = B \cap C?$$

Bài tập 18

Có thể nói gì về các tập A, B nếu $A \Delta B = A$?

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

24

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Tập hợp

Các hằng đẳng thức tập hợp



Bài tập 19

Với A là tập con của một tập vũ trụ U , chứng minh rằng

(a) $A \Delta U = \bar{A}$

(b) $A \Delta \bar{A} = U$

Bài tập 20

Với hai tập A, B bất kỳ, chứng minh

(a) $A \Delta B = (A \cup B) - (A \cap B)$

(b) $A \Delta B = B \Delta A$

(c) $(A \Delta B) \Delta B = A$

Bài tập 21

Chứng minh hoặc tìm phản ví dụ cho các đẳng thức sau

(a) $A \times (B \cup C) = (A \times C) \cup (B \times C)$

(b) $A \times (B \cap C) = (A \times C) \cap (B \times C)$

trong đó A, B, C là các tập bất kỳ

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

25

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Tập hợp

Tổng quát hóa phép hợp và phép giao



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

26

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

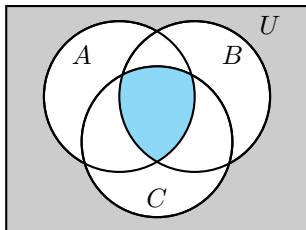
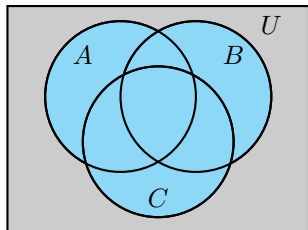
Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- Do các phép hợp và giao thỏa mãn luật giao hoán và luật kết hợp, ta có thể mở rộng các khái niệm này cho dãy n tập $A_1, \dots, A_n$ hoặc thậm chí dãy vô hạn các tập.

- Cách nhóm và thứ tự thực hiện không quan trọng
- $A \cup B \cup C = (A \cup B) \cup C = A \cup (B \cup C) = B \cup (A \cup C) = \dots$
- $A \cap B \cap C = (A \cap B) \cap C = A \cap (B \cap C) = B \cap (A \cap C) = \dots$



Hình: Biểu đồ Venn cho $A \cup B \cup C$ Hình: Biểu đồ Venn cho $A \cap B \cap C$

Tập hợp

Tổng quát hóa phép hợp và phép giao



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

27

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Hợp (union) của một bộ (hữu hạn hoặc vô hạn) các tập hợp là một tập chứa tất cả các phần tử là thành viên của ít nhất một tập trong bộ

$$\bigcup_{i=1}^n A_i = \{x \mid \exists i \in \{1, \dots, n\} (x \in A_i)\} = A_1 \cup A_2 \cup \dots \cup A_n$$

Tương tự với tập chỉ số I bất kỳ $\bigcup_{i \in I} A_i$ hay với vô hạn các

$$\text{tập hợp } \bigcup_{i=1}^{\infty} A_i$$

Ví dụ 5

Với $i = 1, 2, \dots$ nếu $A_i = \{i, i+1, i+2, \dots\}$ thì

$$\bigcup_{i=1}^{\infty} A_i = \bigcup_{i=1}^{\infty} \{i, i+1, i+2, \dots\} = \{1, 2, 3, \dots\} = \mathbb{Z}^+$$

Tập hợp

Tổng quát hóa phép hợp và phép giao



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

28

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Giao (intersection) của một bộ (hữu hạn hoặc vô hạn) các tập hợp là một tập chứa tất cả các phần tử là thành viên của tất cả các tập trong bộ

$$\blacksquare \bigcap_{i=1}^n A_i = \{x \mid \forall i \in \{1, \dots, n\} (x \in A_i)\} = A_1 \cap A_2 \cap \dots \cap A_n$$

■ Tương tự với tập chỉ số I bất kỳ $\bigcap_{i \in I} A_i$ hay với vô hạn các

$$\text{tập hợp } \bigcap_{i=1}^{\infty} A_i$$

Ví dụ 6

Với $i = 1, 2, \dots$ nếu $A_i = \{i, i+1, i+2, \dots\}$ thì

$$\bigcap_{i=1}^n A_i = \bigcap_{i=1}^n \{i, i+1, i+2, \dots\} = \{n, n+1, n+2, \dots\} = A_n$$

Tập hợp

Tổng quát hóa phép hợp và phép giao



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Bài tập 22

Với các tập hợp $A = \{1, 2, 3, 4, 5\}$, $B = \{1, 4, 6, 8\}$, và $C = \{5, 7, 9, 10\}$, tìm $A \cap B \cap C$ và $A \cup B \cup C$

Bài tập 23

Với các tập hợp A, B, C bất kỳ, chứng minh

(a) $\overline{A \cap B \cap C} = \overline{A} \cup \overline{B} \cup \overline{C}$

(b) $\overline{A \cup B \cup C} = \overline{A} \cap \overline{B} \cap \overline{C}$

29

78

Tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

30

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Đây

Định nghĩa dây và một số khái niệm

Một số dây đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- Giả sử **tập vũ trụ U là hữu hạn** và các phần tử của U được liệt kê theo **thứ tự** $u_1, u_2, u_3, \dots, u_n$. Ta có thể biểu diễn một tập hữu hạn $A \subseteq U$ dưới dạng một chuỗi nhị phân $B(A) = x_1 x_2 \dots x_n$ trong đó **$x_i = 1$ nếu $u_i \in A$ và $x_i = 0$ nếu $u_i \notin A$** .

- Với $U = \{1, 2, \dots, 10\}$ ($u_1 = 1, \dots, u_{10} = 10$) và $A = \{2, 3, 5, 7\}$ thì $B(A) = 0110101000$

U	1	2	3	4	5	6	7	8	9	10
$B(A)$	0	1	1	0	1	0	1	0	0	0

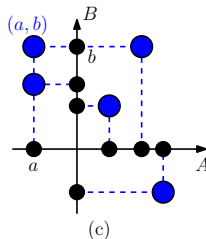
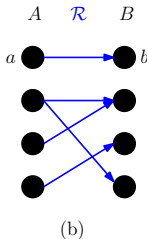
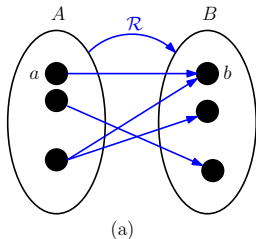
- Các toán tử tập hợp **“ $\cup$ ”, “ $\cap$ ”, và “ $\bar{}$ ”** lần lượt tương ứng với các toán tử logic **“ $\vee$ ”, “ $\wedge$ ”, và “ $\neg$ ”** thực hiện theo từng bit.

Bài tập 24

Với $U = \{1, 2, \dots, 10\}$ ($u_i = i$), $A_1 = \{2, 3, 5, 7\}$, $A_2 = \{1, 3, 9\}$, hãy so sánh

- (1) $B(A_1 \cup A_2)$ và $B(A_1) \vee B(A_2)$
- (2) $B(A_1 \cap A_2)$ và $B(A_1) \wedge B(A_2)$
- (3) $B(\overline{A_1})$ và $\neg B(A_1)$

- Cho hai tập hợp A và B . Một **quan hệ (relation)** $\mathcal{R}$ giữa A và B là một tập con của tích Đề các $A \times B$. Ta viết $a\mathcal{R}b$ nếu $(a, b) \in \mathcal{R}$. Trong trường hợp $A = B$ thì $\mathcal{R}$ được gọi là một quan hệ trong A
- A là tập các giảng viên. B là tập các lớp. $\mathcal{R} \subseteq A \times B$ là quan hệ “phân công giảng viên dạy lớp học”
- $\mathcal{R} = \emptyset$: không có giảng viên nào dạy bất kỳ lớp nào
- $\mathcal{R} = A \times B$: mỗi giảng viên dạy tất cả các lớp
- Biểu diễn một quan hệ bằng hình vẽ



Hình: (a) tương tự giản đồ Venn, (b) đồ thị, (c) hệ tọa độ Đề các



- Một quan hệ $\mathcal{R}$ trong A được gọi là **quan hệ tương đương** (*equivalence relation*) nếu nó thỏa mãn các điều kiện sau
 - Tính phản xạ (*reflexive*) Với mọi a thuộc A , ta có $a\mathcal{R}a$
 - Tính đối xứng (*symmetric*) Với mọi a, b thuộc A , nếu ta có $a\mathcal{R}b$ thì ta cũng có $b\mathcal{R}a$
 - Tính bắc cầu (*transitive*) Với mọi a, b, c thuộc A , nếu ta có $a\mathcal{R}b$ và $b\mathcal{R}c$ thì ta cũng có $a\mathcal{R}c$

Bài tập 25

Trong mỗi trường hợp sau, $\mathcal{R}$ có phải là quan hệ tương đương hay không?

- (1) $\mathcal{R} = \{(p, q) \mid p \equiv q\}$ với p, q là các mệnh đề logic
- (2) $\mathcal{R} = \{(A, B) \mid A \subseteq B\}$ với A, B là các tập hợp
- (3) $\mathcal{R} = \{(A, B) \mid A = B\}$ với A, B là các tập hợp
- (4) $\mathcal{R} = \{(a, b) \mid b \text{ chia hết cho } a\}$ với a, b là các số nguyên dương

Hàm

Định nghĩa hàm và một số khái niệm



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính

chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng

chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số
khái niệm

Một số hàm và toán tử

Đãy

Định nghĩa dãy và một số
khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái
niệm

Một số công thức tổng hữu
ích

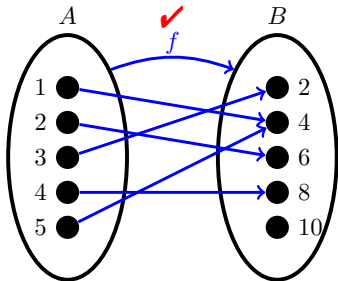
Ký hiệu tích

- Với hai tập khác rỗng A, B , một **hàm (function)** f từ A đến B , ký hiệu $f : A \rightarrow B$, là một quan hệ giữa A và B gán **chính xác một phần tử của B** cho **mỗi phần tử của A**

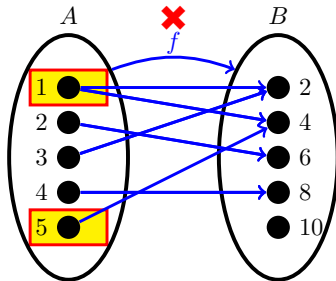
(1) Với mọi $a \in A$, tồn tại $b \in B$ sao cho $(a, b) \in f$

(2) Với b_1 và b_2 thuộc B sao cho $(a, b_1) \in f$ và $(a, b_2) \in f$, ta có $b_1 = b_2$

Nếu b là phần tử duy nhất thuộc B được gán cho phần tử a thuộc A bởi f , ta viết $f(a) = b$



Hình: Hàm



Hình: Không phải hàm

33

78

Hàm

Định nghĩa hàm và một số khái niệm



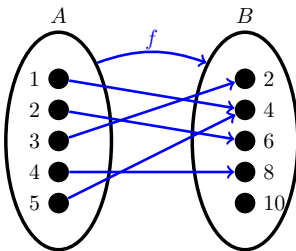
Giả sử f là một hàm từ A đến B

- A được gọi là **miền xác định (domain)** của f
- B được gọi là **miền giá trị (codomain)** của f
- Nếu $f(a) = b$, ta gọi b là **ảnh (image)** của a và a là một **ngược ảnh (preimage)** của b
- Ta cũng nói rằng f **ánh xạ** A đến B

Ví dụ 7

Với hàm f cho bởi hình bên

- Tập xác định $A = \{1, 2, 3, 4, 5\}$
- Tập giá trị $B = \{2, 4, 6, 8, 10\}$
- $4 \in B$ là ảnh của cả $1 \in A$ và $5 \in A$



Hình: $f : A \rightarrow B$

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

34

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Hàm

Định nghĩa hàm và một số khái niệm



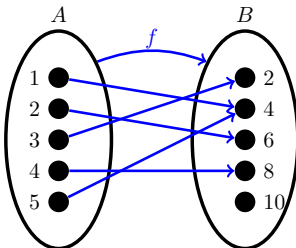
Giả sử f là hàm từ A đến B

- Tập hợp tất cả các ảnh của các phần tử thuộc A được gọi là **ảnh của A qua hàm f** , ký hiệu $f(A)$
 - $f(A) \subseteq B$
- Với tập con $S \subseteq A$, **ảnh của S qua hàm f** , ký hiệu $f(S)$, là tập tất cả các ảnh của các phần tử thuộc S
 - $f(S) = \{t \mid \exists s \in S (t = f(s))\} = \{f(s) \mid s \in S\}$
 - **Chú ý:** $f(s)$ là một phần tử của B và $f(S)$ là một tập con của B

Ví dụ 8

Với hàm f cho bởi hình bên

- $f(A) = \{2, 4, 6, 8\}$
- Với $S = \{1, 2, 5\}$, ta có $f(S) = \{4, 6\}$



Hình: $f : A \rightarrow B$

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

35

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Hàm

Hàm tổng và hàm tích của hai hàm thực



Cho f_1 và f_2 là các hàm từ A đến $\mathbb{R}$. Ta định nghĩa $f_1 + f_2$ và $f_1 f_2$ là các hàm từ A đến $\mathbb{R}$, gọi là các **hàm thực (real-valued function)**, như sau. Với mọi $x \in A$,

ký hiệu hàm

$$\begin{aligned}(f_1 + f_2)(x) &= f_1(x) + f_2(x) \\ (f_1 f_2)(x) &= f_1(x) f_2(x)\end{aligned}$$

phép toán trong $\mathbb{R}$

Bài tập 26

Hãy kiểm tra lại rằng $f_1 + f_2$ và $f_1 f_2$ thực sự là các hàm

Bài tập 27

Gọi F là tập hợp tất cả các hàm $f : \mathbb{R} \rightarrow \mathbb{R}$ với tập xác định và tập giá trị là tập các số thực. Các mệnh đề sau là đúng hay sai? Hãy giải thích đáp án của bạn

(a) $\forall c \in \mathbb{R} [\exists f \in F (f(0) = c)]$

(b) $\exists f \in F [\forall c \in \mathbb{R} (f(0) = c)]$

(c) $\exists f \in F [\forall c \in \mathbb{R} (f(c) = 0)]$

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

36

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

78

Hàm

Hàm hợp

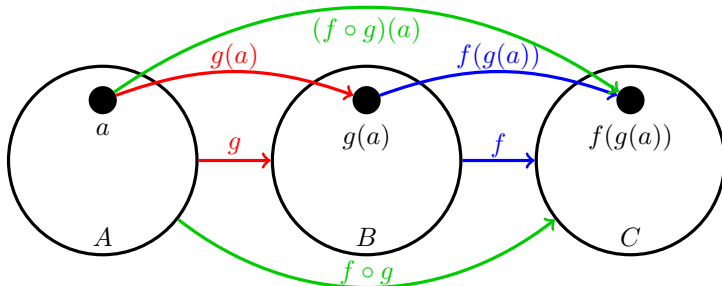


- Với các hàm $g : A \rightarrow B$ và $f : B \rightarrow C$, ta có thể định nghĩa **hợp (composition)** của f và g , ký hiệu $f \circ g : A \rightarrow C$, như sau

$$(f \circ g)(x) = f(g(x))$$

với mọi $x \in A$

- Chú ý:** $f \circ g$ chỉ được định nghĩa khi **tập giá trị của g là tập con của tập xác định của f**
- Chú ý:** Toán tử “ $\circ$ ” không giao hoán, nghĩa là, **trong hầu hết mọi trường hợp, $f \circ g \neq g \circ f$**



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản
Các phép toán trên tập hợp
Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ
Định nghĩa hàm và một số khái niệm
Một số hàm và toán tử

37

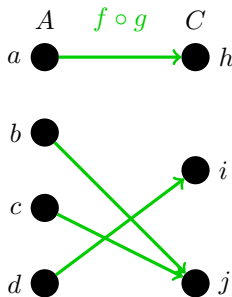
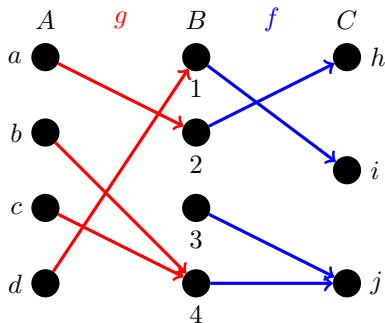
Dãy

Định nghĩa dãy và một số khái niệm
Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm
Một số công thức tổng hữu ích
Ký hiệu tích

Ví dụ 9



Bài tập 28

Cho $g : \{a, b, c\} \rightarrow \{a, b, c\}$ với $g(a) = b, g(b) = c$, và $g(c) = a$.

Cho $f : \{a, b, c\} \rightarrow \{1, 2, 3\}$ với $f(a) = 3, f(b) = 2$, và $f(c) = 1$.

Hãy tìm $f \circ g$ và $g \circ f$

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

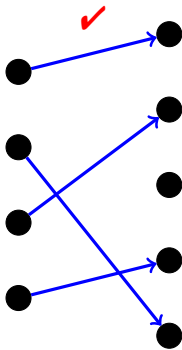
38

78

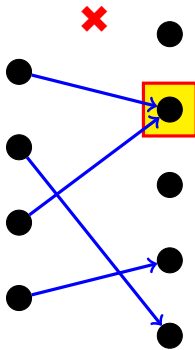
- Hàm $f : A \rightarrow B$ được gọi là một **đơn ánh (injection)** hay một **hàm một-một (one-to-one function)** khi và chỉ khi $f(a) = f(b)$ kéo theo $a = b$ với mọi a và b thuộc tập xác định A của f

■ $\forall a, b (f(a) = f(b) \rightarrow a = b) \equiv \forall a, b (a \neq b \rightarrow f(a) \neq f(b))$

Ví dụ 10



Hình: Đơn ánh



Hình: Không phải đơn ánh

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

40

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Bài tập 29

Hàm $f : \mathbb{Z} \rightarrow \mathbb{Z}$ trong mỗi trường hợp sau đây có phải là đơn ánh không?

(a) $f(n) = n - 1$

(c) $f(n) = n^3$

(b) $f(n) = n^2 + 1$

(d) $f(n) = \lceil n/2 \rceil$



Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

41 Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Cho $f : A \rightarrow B$ là một hàm trong đó A, B là các tập con của $\mathbb{R}$

- f được gọi là **tăng (increasing)** khi và chỉ khi với mọi x, y thuộc A thỏa mãn $x < y$, ta luôn có $f(x) \leq f(y)$

$$\blacksquare \forall x \forall y (x < y \rightarrow f(x) \leq f(y))$$

- f được gọi là **thực sự tăng (strictly increasing)** khi và chỉ khi với mọi x, y thuộc A thỏa mãn $x < y$, ta luôn có $f(x) < f(y)$

$$\blacksquare \forall x \forall y (x < y \rightarrow f(x) < f(y))$$

- f được gọi là **giảm (decreasing)** khi và chỉ khi với mọi x, y thuộc A thỏa mãn $x < y$, ta luôn có $f(x) \geq f(y)$

$$\blacksquare \forall x \forall y (x < y \rightarrow f(x) \geq f(y))$$

- f được gọi là **thực sự giảm (strictly decreasing)** khi và chỉ khi với mọi x, y thuộc A thỏa mãn $x < y$, ta luôn có $f(x) > f(y)$

$$\blacksquare \forall x \forall y (x < y \rightarrow f(x) > f(y))$$



Bài tập 30

- (a) Cho ví dụ về một hàm f là hàm thực sự giảm
- (b) Cho ví dụ về một hàm f là hàm giảm nhưng không là thực sự giảm
- (c) Chứng minh nếu f là hàm thực sự tăng hoặc thực sự giảm thì f là đơn ánh

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

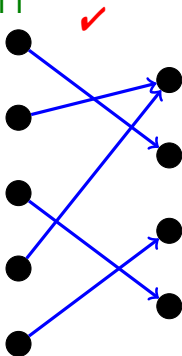
42

Hàm $f : A \rightarrow B$ được gọi là một **toàn ánh (surjection)** khi và chỉ khi với mọi phần tử b thuộc B tồn tại một phần tử a thuộc A sao cho $f(a) = b$

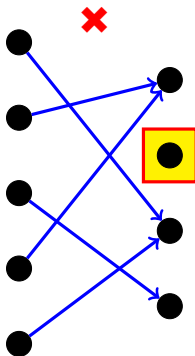
■ $\forall b \in B \exists a \in A (f(a) = b)$

■ $f(A) = B$ (ảnh của A qua f bằng với tập giá trị B)

Ví dụ 11



Hình: Toàn ánh



Hình: Không phải toàn ánh

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

43

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích



Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

44

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Bài tập 31

Hàm $f : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$ trong mỗi trường hợp sau đây có phải là toàn ánh không?

(a) $f(m, n) = 2m - n$

(c) $f(m, n) = m + n + 1$

(b) $f(m, n) = m^2 - n^2$

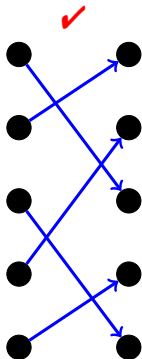
(d) $f(m, n) = m^2 - 4$

Bài tập 32

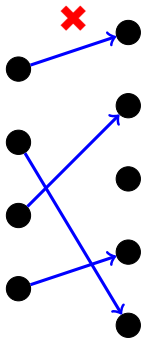
Chứng minh rằng $f : A \rightarrow B$ là toàn ánh khi và chỉ khi $f(A) = B$

Hàm $f : A \rightarrow B$ được gọi là một **song ánh (bijection)** khi và chỉ khi nó đồng thời là đơn ánh và toàn ánh

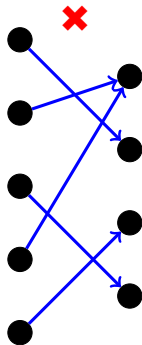
Ví dụ 12



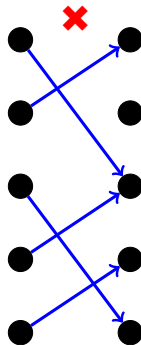
Hình: Song ánh



Hình: Đơn ánh,
không toàn ánh



Hình: Toàn ánh,
không đơn ánh



Hình: Không đơn
ánh, không toàn
ánh

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

45

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

46 Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Chú ý

Với các tập hợp hữu hạn A, B , ta có $|A| = |B|$ khi và chỉ khi tồn tại một song ánh $f : A \rightarrow B$

Bài tập 33

Hàm $f : \mathbb{R} \rightarrow \mathbb{R}$ trong mỗi trường hợp sau đây có phải là song ánh không?

(a) $f(x) = -3x + 4$

(b) $f(x) = -3x^2 + 7$

(c) $f(x) = (x + 1)/(x + 2)$

(d) $f(x) = x^5 + 1$

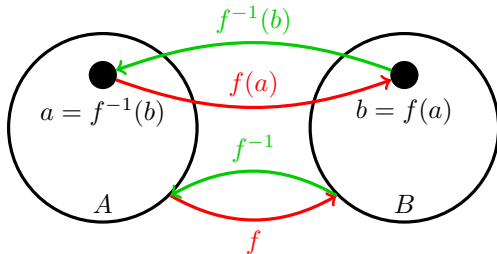
(e) $f(x) = 2x + 1$

(f) $f(x) = x^2 + 1$

(g) $f(x) = x^3$

(h) $f(x) = (x^2 + 1)/(x^2) + 2$

- Cho $f : A \rightarrow B$ là một song ánh. **Hàm ngược (inverse function)** của f là một hàm gán cho mỗi phần tử $b \in B$ một phần tử duy nhất $a \in A$ sao cho $f(a) = b$. Hàm ngược của f được ký hiệu là $f^{-1} : B \rightarrow A$
- Một song ánh còn được gọi là một **hàm khả nghịch (invertible function)**





Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

48

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Bài tập 34

Chứng minh rằng f^{-1} là một song ánh

Bài tập 35

Hàm ngược của các hàm sau có tồn tại hay không? Tại sao?

(a) $f : \mathbb{R} \rightarrow \mathbb{R}, f(x) = x + 1$

(b) $f : \mathbb{R} \rightarrow \mathbb{R}, f(x) = x^2$

(c) $f : \mathbb{R} \rightarrow \mathbb{R}, f(x) = 2x$

(d) $f : \mathbb{N} \rightarrow \mathbb{N}, f(x) = 2x$

Hàm

Hàm đồng nhất



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

49

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

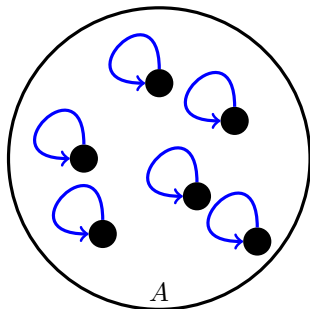
Một số công thức tổng hữu ích

Ký hiệu tích

78

- Cho A là một tập hợp. **Hàm đồng nhất (identity function)** trên A là hàm $\text{id}_A : A \rightarrow A$ trong đó $\text{id}_A(x) = x$ với mọi $x \in A$
- id_A là song ánh với mọi tập A
- Với song ánh $f : A \rightarrow B$ và hàm ngược của nó $f^{-1} : B \rightarrow A$

$$f^{-1} \circ f = \text{id}_A$$



Hình: Hàm đồng nhất trên A



Bài tập 36

Hãy tìm ví dụ một hàm $f : \mathbb{N} \rightarrow \mathbb{N}$ thỏa mãn

- (a) f là đơn ánh nhưng không là toàn ánh
- (b) f là toàn ánh nhưng không là đơn ánh
- (c) f là song ánh và f khác hàm đồng nhất trên $\mathbb{N}$
- (d) f vừa không là đơn ánh vừa không là toàn ánh

Bài tập 37

Cho các hàm $g : A \rightarrow B$ và $f : B \rightarrow C$. Chứng minh rằng

- (a) Nếu cả g và f đều là đơn ánh thì $f \circ g$ cũng là đơn ánh.
- (b) Nếu cả g và f đều là toàn ánh thì $f \circ g$ cũng là toàn ánh.
- (c) Nếu $f \circ g$ là toàn ánh thì f cũng là toàn ánh
- (d) Nếu $f \circ g$ là đơn ánh thì g cũng là đơn ánh
- (e) Nếu $f \circ g$ là song ánh thì g là toàn ánh khi và chỉ khi f là đơn ánh

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản
Các phép toán trên tập hợp
Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ
Định nghĩa hàm và một số khái niệm

50

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm
Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm
Một số công thức tổng hữu ích
Ký hiệu tích



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

51

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Bài tập 38

Tìm ví dụ các hàm f và g thỏa mãn $f \circ g$ là song ánh, nhưng g không phải toàn ánh và f không phải đơn ánh.

Bài tập 39

Gọi $f : A \rightarrow B$ là một hàm với A, B là các tập hữu hạn thỏa mãn $|A| = |B|$. Chứng minh rằng f là đơn ánh khi và chỉ khi nó là toàn ánh.

Hàm

Hàm sàn và hàm trần



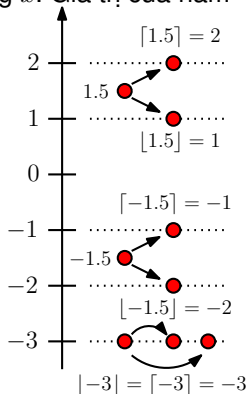
Trong toán rời rạc, ta thường dùng hai hàm sau

- **Hàm sàn (floor function)** gán cho số thực x số nguyên lớn nhất có giá trị nhỏ hơn hoặc bằng x . Giá trị của hàm sàn được ký hiệu là $\lfloor x \rfloor$
- **Hàm trần (ceiling function)** gán cho số thực x số nguyên nhỏ nhất có giá trị lớn hơn hoặc bằng x . Giá trị của hàm trần được ký hiệu là $\lceil x \rceil$

- Nếu $x \notin \mathbb{Z}$ thì $\lfloor -x \rfloor \neq -\lfloor x \rfloor$ và $\lceil -x \rceil \neq -\lceil x \rceil$
- Nếu $x \in \mathbb{Z}$ thì $\lfloor x \rfloor = \lceil x \rceil = x$

Ví dụ 13

- $\lfloor 1.5 \rfloor = 1, \lceil 1.5 \rceil = 2$
- $\lfloor -1.5 \rfloor = -2, \lceil -1.5 \rceil = -1$
- $\lfloor -3 \rfloor = -3, \lceil -3 \rceil = -3$



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

52

78

Hàm

Hàm sàn và hàm trần



Các cấu trúc cơ bản

Hoàng Anh Đức

Bài tập 40

Chứng minh các tính chất sau của hàm trần và hàm sàn, trong đó $x \in \mathbb{R}$ và $n \in \mathbb{Z}$

$$(1a) \lfloor x \rfloor = n \text{ khi và chỉ khi } n \leq x < n + 1$$

$$(1b) \lceil x \rceil = n \text{ khi và chỉ khi } n - 1 < x \leq n$$

$$(1c) \lfloor x \rfloor = n \text{ khi và chỉ khi } x - 1 < n \leq x$$

$$(1d) \lceil x \rceil = n \text{ khi và chỉ khi } x \leq n < x + 1$$

$$(2) x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$$

$$(3a) \lfloor -x \rfloor = -\lceil x \rceil$$

$$(3b) \lceil -x \rceil = -\lfloor x \rfloor$$

$$(4a) \lfloor x + n \rfloor = \lfloor x \rfloor + n$$

$$(4b) \lceil x + n \rceil = \lceil x \rceil + n$$

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

53

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Hàm

Hàm sàn và hàm trần



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

54

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Bài tập 41

Chứng minh rằng nếu $n \in \mathbb{N}$ thì $\lfloor n/2 \rfloor = n/2$ nếu n chẵn và $\lfloor n/2 \rfloor = (n-1)/2$ nếu n lẻ

Bài tập 42

Chứng minh rằng nếu $x \in \mathbb{R}$ thì $\lfloor 2x \rfloor = \lfloor x \rfloor + \lfloor x + 1/2 \rfloor$. (Gợi ý: Khi xét các bài toán liên quan đến hàm sàn, một cách tiếp cận hữu ích là đặt $x = n + \epsilon$ trong đó $n = \lfloor x \rfloor \in \mathbb{Z}$ và ϵ là một số thực thỏa mãn $0 \leq \epsilon < 1$. Tương tự, với hàm trần, có thể đặt $x = n - \epsilon$)

Dãy

Định nghĩa dãy và một số khái niệm



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

55

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Một *dãy (sequence)* là một cấu trúc rời rạc được dùng để biểu diễn một *danh sách có thứ tự*

Ví dụ 14

- 1, 2, 3, 5, 8 là một dãy gồm năm số hạng
- 1, 3, 9, 27, 81, ..., 3^n , ... là một dãy vô hạn

Dãy

Định nghĩa dãy và một số khái niệm



Dãy

Một **dãy (sequence)** là một hàm $f : I \rightarrow S$ trong đó $I \subseteq \mathbb{Z}$ và S là một tập hợp bất kỳ

- Tập I được gọi là **tập các chỉ số (indices)** của dãy. Thông thường $I = \mathbb{N}$ hoặc $I = \mathbb{Z}^+$
- Với số nguyên $i \in I$, ta ký hiệu $a_i = f(i)$ là ảnh của i qua f , và gọi a_i là **số hạng (term)** hoặc **phần tử (element)** của dãy xác định bởi f và i là **chỉ số (index)** của số hạng a_i
- Ta sử dụng ký hiệu $\{a_n\}$ để chỉ dãy xác định bởi f và nói rằng $\{a_n\}$ là dãy cho bởi $a_n = f(n)$
 - Một số ký hiệu khác là $\{a_n\}_{n \in I}$, $(a_n)_{n \in I}$, hoặc $(f(n))_{n \in I}$
 - Có thể mô tả một dãy bằng cách liệt kê một vài số hạng đầu tiên hoặc cuối cùng của dãy và sử dụng “...” cho phần còn lại
 - Chữ a trong ký hiệu dãy có thể được thay bằng các chữ cái khác
 - Lưu ý rằng ký hiệu $\{a_n\}$ dùng để chỉ một dãy có thể xung đột với ký hiệu dành cho một tập hợp. Tuy nhiên, tùy vào ngữ cảnh sử dụng ký hiệu này, ta sẽ luôn phân biệt được rõ khi nào ta đang xét một tập hợp và khi nào ta đang xét một dãy.

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

56

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Dãy

Định nghĩa dãy và một số khái niệm



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

57

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Ví dụ 15

Dãy $\{a_n\}$ xác định bởi $f : \mathbb{N} \rightarrow \mathbb{Z}$ với $f(n) = n^2$

- Các số hạng của dãy là $0, 1, 4, 9, 16, 25, \dots$ (nghĩa là $a_0 = 0, a_1 = 1, a_2 = 4, a_3 = 9, a_4 = 16, a_5 = 25, \dots$)
- Có thể viết $\{a_n\} = 0, 1, 4, 9, 16, 25, \dots$
- $\{a_n\}$ là dãy cho bởi $a_n = n^2$

Bài tập 43

Trong mỗi trường hợp sau, tìm các số hạng $a_0, a_1, \dots, a_5$ của dãy $\{a_n\}$ với

(a) $a_n = 2^{n-1}$

(b) $a_n = 1 + (-1)^n$

(c) $a_n = (n+1)^{n+1}$

(d) $a_n = 7$

(e) $a_n = -(-2)^n$

(f) $a_n = \lfloor n/2 \rfloor$

Dãy

Định nghĩa dãy và một số khái niệm



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

58

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- Một **chuỗi ký tự (string)** có thể được coi là một dãy hữu hạn các ký tự viết liền nhau, ví dụ như $a_1 a_2 \dots a_n$
- Độ dài của chuỗi ký tự là số ký tự trong chuỗi
- Chuỗi ký tự rỗng λ là chuỗi không có ký tự nào (độ dài bằng 0)

Dãy

Cấp số nhân và cấp số cộng



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- Một **cấp số nhân (geometric progression)** là một dãy có dạng

$$a, ar, ar^2, \dots, ar^n, \dots$$

trong đó **số hạng đầu tiên (initial term)** a và **công bội (common ratio)** r là các số thực

- Ví dụ, với $n = 0, 1, 2, \dots$

- $\{b_n\}$ với $b_n = (-1)^n$

số hạng đầu tiên 1, công bội -1

- $\{c_n\}$ với $c_n = 6 \cdot (1/3)^n$

số hạng đầu tiên 6, công bội $1/3$

- Một **cấp số cộng (arithmetic progression)** là một dãy có dạng

$$a, a + d, a + 2d, \dots, a + nd, \dots$$

trong đó **số hạng đầu tiên (initial term)** a và **công sai (common difference)** d là các số thực

- Ví dụ, với $n = 0, 1, 2, \dots$

- $\{d_n\}$ với $d_n = -1 + 4n$

số hạng đầu tiên -1 , công sai 4

- $\{e_n\}$ với $e_n = 7 - 3n$

số hạng đầu tiên 7, công sai -3

59

78

Dãy

Tìm công thức tường minh của một dãy



Bài toán

- Cho trước một vài phần tử của dãy
- Yêu cầu tìm
 - một công thức tường minh của các số hạng
 - hoặc một phương thức để liệt kê các phần tử của dãy

Ví dụ 16

Số tiếp theo trong dãy có thể là bao nhiêu?

- 1, 2, 3, 4, ...
- 1, 3, 5, 7, 9, ...
- 2, 3, 5, 7, 11, ...

Ví dụ 17

Các số hạng tiếp theo có thể là bao nhiêu?

- 1, 2, 2, 3, 3, 3, 4, 4, 4, 4
- 0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

60

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

78

Dãy

Tìm công thức tường minh của một dãy



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- Một phương pháp hữu ích để tìm công thức tổng quát cho các số hạng của một dãy là *so sánh các số hạng của dãy cần tìm với các số hạng của một dãy đã biết* (ví dụ như cấp số cộng, cấp số nhân, dãy số chính phương, v.v...)

Công thức	Mười số hạng đầu tiên
n^2	1, 4, 9, 16, 25, 36, 49, 64, 81, 100, ...
n^3	1, 8, 27, 64, 125, 216, 343, 512, 729, 1000, ...
n^4	1, 16, 81, 256, 625, 1296, 2401, 4096, 6561, 10000, ...
f_n	1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...
2^n	2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, ...
3^n	3, 9, 27, 81, 243, 729, 2187, 6561, 19683, 59049, ...
$n!$	1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800, ...

- Bảng Tra Cứu Dãy Số Nguyên Trực Tuyến (The On-Line Encyclopedia of Integer Sequences - OEIS)

<https://oeis.org/>

61

78

Dãy

Tìm công thức tường minh của một dãy



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

62

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Bài tập 44

Với mỗi dãy số nguyên sau, hãy tìm một công thức đơn giản hoặc một cách để tìm các số hạng tiếp theo của dãy. Giả sử công thức bạn tìm ra là đúng, hãy tìm ba số hạng tiếp theo của dãy

(a) $1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, \dots$

(b) $1, 2, 2, 3, 4, 4, 5, 6, 6, 7, 8, 8, \dots$

(c) $1, 0, 2, 0, 4, 0, 8, 0, 16, 0, \dots$

(d) $3, 6, 12, 24, 48, 96, 192, \dots$

(e) $15, 8, 1, -6, -13, -20, -27, \dots$

(f) $3, 5, 8, 12, 17, 23, 30, 38, 47, \dots$

(g) $2, 16, 54, 128, 250, 432, 686, \dots$

(h) $2, 3, 7, 25, 121, 721, 5041, 40321, \dots$

Tổng

Ký hiệu tổng và một số khái niệm



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

63

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- Ta giới thiệu **ký hiệu tổng (summation notation)** để biểu diễn tổng của các số hạng trong một dãy
- Cho **dãy** $\{a_n\}$, một số nguyên **giới hạn dưới (lower limit)** m , và một số nguyên **giới hạn trên (upper limit)** $n \geq m$. **Tổng (summation)** của các số hạng $a_m, a_{m+1}, \dots, a_n$ có thể được viết là

$$a_m + a_{m+1} + \dots + a_n \quad \text{hoặc}$$

$$\sum_{j=m}^n a_j \quad \text{hoặc} \quad \sum_{m \leq j \leq n} a_j$$

- Ở đây, j được gọi là **chỉ số lấy tổng (index of summation)** và được chọn hoàn toàn tùy ý

$$\sum_{j=m}^n a_j = \sum_{i=m}^n a_i = \sum_{k=m}^n a_k$$

- Để thuận tiện, ta quy ước $\sum_{i=m}^n a_i = 0$ với mọi $m > n$

Tổng

Ký hiệu tổng và một số khái niệm



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

64

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Ví dụ 18

Cho dãy $\{a_n\}$,

- Với tập chỉ số S bất kỳ, tổng các số hạng a_i với $i \in S$:

$$\sum_{i \in S} a_i$$

- Tổng các số hạng lớn hơn hoặc bằng a_j :

$$\sum_{i=j}^{\infty} a_i = a_j + a_{j+1} + \dots$$

Tổng

Ký hiệu tổng và một số khái niệm



Ví dụ 19

Tổng đôi (double sum) xuất hiện trong nhiều ngữ cảnh (ví dụ khi phân tích các vòng lồng nhau trong chương trình máy tính).

Ví dụ:

$$\sum_{i=1}^4 \sum_{j=1}^3 ij.$$

Để tính tổng này, khai triển tổng bên trong trước và sau đó tính tổng bên ngoài:

$$\begin{aligned} \sum_{i=1}^4 \sum_{j=1}^3 ij &= \sum_{i=1}^4 \left(\sum_{j=1}^3 ij \right) = \sum_{i=1}^4 (i(1 + 2 + 3)) = \sum_{i=1}^4 6i \\ &= 6 \sum_{i=1}^4 i = 6 \cdot \frac{4 \cdot 5}{2} = 60. \end{aligned}$$

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

65

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Tổng

Ký hiệu tổng và một số khái niệm



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

66

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Ví dụ 20

Với hàm $f : X \rightarrow \mathbb{R}$,

- Với $X = \{x_1, x_2, \dots\}$, tổng các giá trị của hàm f :

$$\sum_{x \in X} f(x) = f(x_1) + f(x_2) + \dots$$

- Với $X = \{x \mid P(x)\}$ trong đó $P(x)$ là vị từ cho trước, tổng các giá trị của hàm f :

$$\sum_{P(x)} f(x) = f(x_1) + f(x_2) + \dots$$

Tổng

Ký hiệu tổng và một số khái niệm



Ví dụ 21

$$\sum_{j=1}^4 j^2 = 1^2 + 2^2 + 3^2 + 4^2$$

$$\sum_{j=1}^{100} \frac{1}{j} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{100}$$

$$\sum_{j=0}^{\infty} 2^j = 1 + 2 + 2^2 + 2^3 + \dots$$

$$\sum_{\substack{(0 \leq x \leq 10) \\ \wedge (x \text{ chẵn})}} x^2 = 0 + 2^2 + 4^2 + 6^2 + 8^2 + 10^2$$

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

67

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

78

Tổng

Ký hiệu tổng và một số khái niệm



Bài tập 45

Sử dụng ký hiệu tổng để viết lại các công thức sau

(a) $2 + 4 + 6 + 8 + \cdots + 2n$

(b) $1 + 5 + 9 + 13 + \cdots + 425$

(c) $1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{50}$

Bài tập 46

Viết lại các tổng sau dưới dạng công thức dài hơn

(a) $\sum_{k=1}^{100} (3 + 4k)$

(b) $\sum_{k=2}^{50} \frac{1}{k^2 - 1}$

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

68

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

78

Tổng

Một số công thức tổng hữu ích



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- **Tổng hằng số:** Với hằng số c bất kỳ,

$$\sum_{n=i}^j c = (j - i + 1) \cdot c$$

- **Phân phối:** Với hằng số c bất kỳ,

$$\sum_{n=i}^j c f(n) = c \sum_{n=i}^j f(n)$$

- **Giao hoán:**

$$\sum_{n=i}^j (f(n) + g(n)) = \sum_{n=i}^j f(n) + \sum_{n=i}^j g(n)$$

Tổng

Một số công thức tổng hữu ích



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

■ Đổi chỉ số:

$$\sum_{i=j}^m f(i) = \sum_{k=j+n}^{m+n} f(k-n)$$

■ Ví dụ $\sum_{i=1}^4 i^2 = \sum_{k=3}^6 (k-2)^2$ (đặt $k = i + 2$)

■ Tách tổng: Với $j \leq m < k$

$$\sum_{i=j}^k f(i) = \sum_{i=j}^m f(i) + \sum_{i=m+1}^k f(i)$$

■ Đảo thứ tự:

$$\sum_{i=0}^k f(i) = \sum_{i=0}^k f(k-i)$$

70

78

Tổng

Một số công thức tổng hữu ích



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

- Với $\{a_n\}$ là cấp số nhân có số hạng đầu tiên a và công bội r , tổng của $n + 1$ số hạng đầu tiên của dãy là

$$S = \sum_{i=0}^n ar^i$$

- Công thức tường minh

$$S = \sum_{i=0}^n ar^i = \begin{cases} \frac{ar^{n+1} - a}{r - 1} & \text{nếu } r \neq 1 \\ (n + 1)a & \text{nếu } r = 1 \end{cases}$$

$$S = a + ar + ar^2 + ar^3 + \dots + ar^n$$

$$rS = ar + ar^2 + ar^3 + \dots + ar^n + ar^{n+1}$$

$$rS - S = ar^{n+1} - a$$

71

78

Tổng

Một số công thức tổng hữu ích



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

công thức của S

phân phối

đổi chỉ số, $k = i + 1$

tách tổng

thêm và bớt $ar^0 = a$

tách tổng

công thức của S

$$rS = r \sum_{i=0}^n ar^i$$

$$= \sum_{i=0}^n ar^{i+1}$$

$$= \sum_{k=1}^{n+1} ar^k$$

$$= \sum_{k=1}^n ar^k + \sum_{k=n+1}^{n+1} ar^k$$

$$= \left(\sum_{k=1}^n ar^k + ar^0 \right) + (ar^{n+1} - ar^0)$$

$$= \sum_{k=0}^n ar^k + (ar^{n+1} - a)$$

$$= S + (ar^{n+1} - a)$$

72

78

Tổng

Một số công thức tổng hữu ích



Ví dụ 22

Tìm công thức tường minh cho tổng $T = \sum_{i=1}^n i$

$$T = 1 + 2 + 3 + \cdots + n$$

$$T = n + (n - 1) + (n - 2) + \cdots + 1$$

$$2T = (n + 1) \cdot n$$

Bài tập 47

Với $\{a_n\}$ là cấp số cộng có số hạng đầu tiên a và công sai d , tổng của $n + 1$ số hạng đầu tiên của dãy là

$$T = \sum_{i=0}^n (a + id)$$

Hãy tìm công thức tường minh cho T

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

73

78

Tổng

Một số công thức tổng hữu ích



Ví dụ 23

Phương pháp của Gauss để tính $\sum_{i=1}^{100} i$

Ý tưởng: Ghép số hạng đầu tiên với số hạng cuối cùng, số hạng thứ hai với số hạng áp chót, v.v.

Bài tập 48

Tìm công thức tường minh cho $T = \sum_{i=1}^n i$ sử dụng ý tưởng

tương tự. Có thể áp dụng phương pháp tương tự cho Bài tập 47 không?

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

74

78

Tổng

Một số công thức tổng hữu ích



Các cấu trúc cơ bản

Hoàng Anh Đức

Ví dụ 24

Tìm công thức tường minh của $T = \sum_{n=0}^{\infty} x^n$ với x là số thực thỏa

mãn $-1 < x < 1$

Ta đã chứng minh

$$\sum_{n=0}^k x^n = \frac{x^{k+1} - 1}{x - 1}.$$

Do $-1 < x < 1$, $x^{k+1} \rightarrow 0$ khi $k \rightarrow \infty$. Ta có

$$T = \sum_{n=0}^{\infty} x^n = \lim_{k \rightarrow \infty} \sum_{n=0}^k x^n = \lim_{k \rightarrow \infty} \frac{x^{k+1} - 1}{x - 1} = \frac{1}{1 - x}$$

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

75

78

Tổng

Một số công thức tổng hữu ích



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Tổng	Công thức tường minh
$\sum_{k=0}^n ar^k \ (r \neq 0)$	$\frac{ar^{n+1} - a}{r - 1} \ (r \neq 1)$
$\sum_{k=1}^n k$	$\frac{n(n+1)}{2}$
$\sum_{k=1}^n k^2$	$\frac{n(n+1)(2n+1)}{6}$
$\sum_{k=1}^n k^3$	$\frac{n^2(n+1)^2}{4}$
$\sum_{k=0}^{\infty} x^k \ (-1 < x < 1)$	$\frac{1}{1-x}$
$\sum_{k=1}^{\infty} kx^{k-1} \ (-1 < x < 1)$	$\frac{1}{(1-x)^2}$

Tổng

Một số công thức tổng hữu ích



Bài tập 49

Tính các tổng sau

$$(a) \sum_{k=100}^{200} k \quad (b) \sum_{k=99}^{200} k^3 \quad (c) \sum_{10}^{20} k^2(k-3)$$

Bài tập 50

(a) Chứng minh rằng $\sum_{i=1}^n (a_i - a_{i-1}) = a_n - a_0$, trong đó

$a_0, a_1, \dots, a_n$ là một dãy gồm các số thực

(b) Sử dụng đẳng thức $\frac{1}{k(k+1)} = \frac{1}{k} - \frac{1}{k+1}$ và phần (a) để

$$\text{tính } \sum_{k=1}^n \frac{1}{k(k+1)}$$

Bài tập 51

Lấy tổng cả hai vế của đẳng thức $k^2 - (k-1)^2 = 2k - 1$ từ $k = 1$ đến $k = n$ và sử dụng Bài tập 50(a) để tìm một công thức tường minh cho $\sum_{k=1}^n (2k - 1)$ (tổng n số tự nhiên lẻ đầu tiên)

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

77

78

Tích

Ký hiệu tích



Ngoài ký hiệu tổng, ta cũng có ký hiệu đặc biệt cho **tích (product)**. Cho dãy $\{a_n\}$ và các chỉ số giới hạn dưới m và giới hạn trên $n \geq m$. Tích của các số hạng $a_m, a_{m+1}, \dots, a_n$ có thể được viết là

$$a_m \cdot a_{m+1} \cdot \dots \cdot a_n \quad \text{hoặc}$$

$$\prod_{j=m}^n a_j \quad \text{hoặc} \quad \prod_{m \leq j \leq n} a_j$$

Để thuận tiện, ta quy ước $\prod_{i=m}^n a_i = 1$ với mọi $m > n$

Bài tập 52

Tính giá trị của các tích sau

$$(a) \prod_{i=0}^{10} i$$

$$(b) \prod_{i=1}^{100} (-1)^i$$

$$(c) \prod_{i=5}^8 i$$

$$(d) \prod_{i=1}^{10} 2$$

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Một số khái niệm và tính chất cơ bản

Các phép toán trên tập hợp

Biểu diễn tập hợp bằng chuỗi nhị phân

Hàm

Quan hệ

Định nghĩa hàm và một số khái niệm

Một số hàm và toán tử

Dãy

Định nghĩa dãy và một số khái niệm

Một số dãy đặc biệt

Tổng/Tích

Ký hiệu tổng và một số khái niệm

Một số công thức tổng hữu ích

Ký hiệu tích

Part I

Phụ lục

Nội dung



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

Tập đếm được và không đếm được

Một số lỗi thường gặp

Tập hợp
Nghịch lý

Lực lượng của tập vô hạn
Định nghĩa
Tập đếm được và không đếm được

Một số lỗi thường gặp



- Chúng ta đang học một *lý thuyết tập hợp ngây thơ (naive set theory)*
 - Định nghĩa bằng ngôn ngữ tự nhiên, không chặt chẽ về mặt toán học
 - Mô tả các khía cạnh của các tập hợp toán học quen thuộc trong toán rời rạc
 - Bản thân lý thuyết này có chứa các *nghịch lý (paradox)* (= một phát biểu tự phủ định chính nó mặc dù lúc đầu nhìn có vẻ đúng)
- *Nghịch lý Russell* (Đặt theo tên nhà triết học, nhà logic học, nhà toán học người Anh Bertrand Russell (1872–1970))
 - Gọi S là tập *tất cả các tập hợp không chứa chính nó như là một phần tử*, nghĩa là $S = \{A \mid A \text{ là một tập hợp và } A \notin A\}$
 - Chú ý rằng *theo định nghĩa tập hợp ta đã học, tồn tại một tập hợp chứa chính nó như là một phần tử*. Ví dụ xét *tập T các tập hợp có chứa ít nhất một phần tử*
 - *Liệu S có phải là một phần tử của chính nó hay không*, nói cách khác, liệu $S \in S$?

Lực lượng của tập vô hạn

Định nghĩa



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

3 Định nghĩa

Tập đếm được và không đếm được

Một số lỗi thường gặp

- **Nhắc lại:** *Lực lượng (cardinality)* của một tập A , ký hiệu $|A|$, là số phần tử khác biệt mà A có
- Các tập A và B *có cùng lực lượng*, ký hiệu $|A| = |B|$, khi và chỉ khi tồn tại một *song ánh* từ A đến B
- Nếu tồn tại một *đơn ánh* từ A đến B , ta nói “lực lượng của A nhỏ hơn hoặc bằng lực lượng của B ”, và ký hiệu $|A| \leq |B|$
- Khi $|A| \leq |B|$ và hai tập A, B có lực lượng khác nhau, ta nói “lực lượng của A nhỏ hơn lực lượng của B ”, và ký hiệu $|A| < |B|$

Bài tập 53

Chứng minh rằng $|A| \leq |\mathcal{P}(A)|$ với mọi tập hợp A , trong đó $\mathcal{P}(A)$ là tập tất cả các tập hợp con của A

Bài tập 54

Tập $2\mathbb{Z}$ gồm các số nguyên chẵn có cùng lực lượng với tập số nguyên $\mathbb{Z}$ hay không?

Lực lượng của tập vô hạn

Định nghĩa



Định lý 1: Định lý Cantor

Không tồn tại một toàn ánh $f : A \rightarrow \mathcal{P}(A)$ với A là một tập hợp bất kỳ và $\mathcal{P}(A)$ là tập tất cả các tập con của A

Chứng minh.

Ta chứng minh bằng phương pháp phản chứng

- Giả sử tồn tại toàn ánh $f : A \rightarrow \mathcal{P}(A)$
- Ta định nghĩa tập con $G \subseteq A$ như sau

$$G := \{x \in A \mid x \notin f(x)\}$$

- Do f là toàn ánh, tồn tại $a \in A$ sao cho $G = f(a)$
- Xét hai trường hợp
 - Nếu $a \in G$ thì theo định nghĩa của G , ta có $a \notin f(a) = G$. Đây là một mâu thuẫn
 - Nếu $a \notin G = f(a)$ thì $a \in f(a)$. Do đó theo định nghĩa của G , ta có $a \in G$. Đây là một mâu thuẫn

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

4

Định nghĩa

Tập đếm được và không đếm được

Một số lỗi thường gặp

Lực lượng của tập vô hạn

Tập đếm được và không đếm được



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

Tập đếm được và không đếm được

Một số lỗi thường gặp

- Một tập có hữu hạn số phần tử hoặc có cùng lực lượng với tập các số nguyên dương $\mathbb{Z}^+$ được gọi là **tập đếm được** (*countable set*) và ngược lại thì gọi là **tập không đếm được** (*uncountable set*)
 - Có thể liệt kê các phần tử của tập đếm được theo thứ tự: phần tử thứ 1, phần tử thứ 2, v.v...
- Khi một tập vô hạn S là tập đếm được, ta ký hiệu lực lượng của S là $\aleph_0$ ("aleph null") và viết $|S| = \aleph_0$

Ví dụ 25

Tập các số tự nhiên $\mathbb{N}$ là tập đếm được

0	1	2	3	4	...
↕	↕	↕	↕	↕	
1	2	3	4	5	...

Ví dụ 26

Tập các số nguyên dương lẻ là tập đếm được

1	2	3	4	5	...
↕	↕	↕	↕	↕	
1	3	5	7	9	...

5

17

Lực lượng của tập vô hạn

Tập đếm được và không đếm được



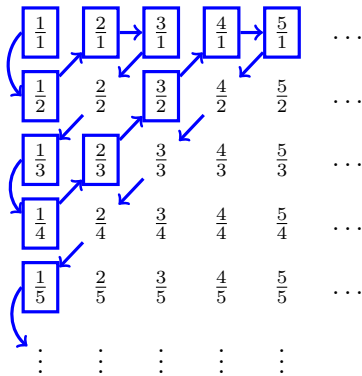
Ví dụ 27

Tập các số nguyên $\mathbb{Z}$ là tập đếm được

$$\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & \dots \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \\ 0 & 1 & -1 & 2 & -2 & \dots \end{array}$$

Ví dụ 28

Tập các số hữu tỷ dương $\mathbb{Q}^+ = \{\frac{p}{q} \mid p, q \in \mathbb{Z}^+\}$ là tập đếm được



$$\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & \dots \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \\ \frac{1}{1} & \frac{1}{2} & \frac{2}{1} & \frac{3}{1} & \frac{1}{3} & \dots \end{array}$$

Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

6

Tập đếm được và không đếm được

Một số lỗi thường gặp

Lực lượng của tập vô hạn

Tập đếm được và không đếm được



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

Tập đếm được và không đếm được

Một số lỗi thường gặp

Ta chứng minh *tập số thực $\mathbb{R}$ là tập không đếm được* bằng phương pháp phản chứng sử dụng *lập luận đường chéo của Cantor (Cantor diagonalization argument)*

- Giả sử $\mathbb{R}$ là tập đếm được. Do mọi tập con của một tập đếm được cũng là một tập đếm được (*tại sao?*), tập các số thực nằm giữa 0 và 1 cũng là tập đếm được
- Sắp thứ tự các số thực giữa 0 và 1: $r_1, r_2, \dots$

$$r_1 = 0.d_{11}d_{12}d_{13}d_{14}\dots$$

$$r_2 = 0.d_{21}d_{22}d_{23}d_{24}\dots$$

$$r_3 = 0.d_{31}d_{32}d_{33}d_{34}\dots$$

$$r_4 = 0.d_{41}d_{42}d_{43}d_{44}\dots$$

$\vdots$

trong đó $d_{ij} \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Lực lượng của tập vô hạn

Tập đếm được và không đếm được



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

8

Tập đếm được và không đếm được

Một số lỗi thường gặp

- Xây dựng một số thực $r = 0.d_1d_2d_3d_4\dots$ mới trong đó

$$d_i = \begin{cases} 4 & \text{nếu } d_{ii} \neq 4 \\ 5 & \text{nếu } d_{ii} = 4 \end{cases}$$

- r không bằng bất cứ số nào trong các số $r_1, r_2, \dots$ vì nó luôn khác r_i ở vị trí thứ i sau “0.”
- Do đó r là một số thực giữa 0 và 1 không nằm trong danh sách $r_1, r_2, \dots$, do mỗi số thực có một biểu diễn thập phân duy nhất
- Tóm lại, không phải mọi số thực giữa 0 và 1 đều được liệt kê theo thứ tự $r_1, r_2, \dots$, và do đó tập các số thực giữa 0 và 1 là tập không đếm được
- Nếu tập con của một tập là không đếm được thì tập đó cũng không đếm được (*tại sao?*), suy ra tập số thực $\mathbb{R}$ là không đếm được

Một số lỗi thường gặp



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

Tập đếm được và không đếm được

Chú ý

Tham khảo từ tài liệu “Common Mistakes in Discrete Mathematics” (https://highered.mheducation.com/sites/dl/free/125967651x/1106131/Common_Mistakes_in_Discrete_Math.pdf)

(a) Lấy phần bù của tập hợp không chính xác khi sử dụng sai luật De Morgan

- Chẳng hạn, cho rằng $\overline{A \cap B} = \overline{A} \cap \overline{B}$
- Các phát biểu đúng là $\overline{A \cap B} = \overline{A} \cup \overline{B}$ và $\overline{A \cup B} = \overline{A} \cap \overline{B}$
- Đây là một trường hợp của việc giả định rằng mọi toán tử phân phối qua toán tử khác, ở đây là phép bù phân phối qua phép giao (hoặc phép hợp). Sinh viên đôi khi mắc lỗi tương tự trong đại số, chẳng hạn như khẳng định, không chính xác, rằng $\sqrt{a^2 + b^2} = a + b$ hoặc $\sin(\alpha + \beta) = \sin \alpha + \sin \beta$.

9 Một số lỗi thường gặp

Một số lỗi thường gặp (tiếp)



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

Tập đếm được và không đếm được

(b) Sử dụng ký hiệu không chính xác về phần tử và tập con của tập hợp lũy thừa và nhầm lẫn giữa khái niệm “phần tử” và “tập con” khi xử lý tập hợp lũy thừa

- Nếu A là tập con của S , thì A là một phần tử của tập lũy thừa của S
- Ví dụ, nếu $S = \{p, q, r\}$, thì $\{p, r\} \subseteq S$, nên $\{p, r\} \in \mathcal{P}(S)$. Mặt khác, $\{p, r\} \not\subseteq \mathcal{P}(S)$, và $\{p, r\} \notin S$. Ngoài ra, lưu ý rằng $\emptyset \notin S$, $\emptyset \in \mathcal{P}(S)$, và $\{\emptyset\} \notin \mathcal{P}(S)$

(c) Viết $\{\emptyset\}$ để biểu diễn tập rỗng

- Một lý do tại sao đây không phải là tập rỗng vì nó có một phần tử trong đó
- Ký hiệu đúng cho tập rỗng là $\{\}$ hoặc $\emptyset$. Tập rỗng là một tập hợp không có phần tử nào

(d) Bỏ sót dấu ngoặc đơn một cách không chính xác trong các biểu thức liên quan đến giao, hợp và hiệu của tập hợp

10

Một số lỗi thường gặp

17

Một số lỗi thường gặp (tiếp)



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

Tập đếm được và không đếm được

(e) Không coi hàm số là một đối tượng ở mức trừu tượng cao hơn so với phần tử, cặp có thứ tự, hoặc tập hợp

- Ký hiệu $f : A \rightarrow B$ có nghĩa là f là một quy trình hay một quy tắc phải áp dụng cho *mọi phần tử của A* và *trong mỗi trường hợp cho ra kết quả trong B* .
- Một hàm không phải là A , không phải là B , không phải là phần tử của A hoặc B , nó cũng không phải là tác động trên chỉ một phần tử của A . Ví dụ, nếu f là hàm số từ $\{1, 2, 3\}$ đến các số tự nhiên có quy tắc $f(x) = x^2$, thì f là toàn bộ quy trình gán 1 cho 1, 2 cho 4, và 3 cho 9; nó không phải là, ví dụ, chỉ số 9 hoặc chỉ là cặp $(3, 9)$.

11 Một số lỗi thường gặp

Một số lỗi thường gặp (tiếp)



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

Tập đếm được và không đếm được

(f) Nhầm lẫn giữa ý tưởng rằng một hàm phải được định nghĩa rõ ràng với khái niệm một-một

- Một hàm nói chung không cần có tính chất là các phần tử khác nhau của miền xác định được ánh xạ đến các phần tử khác nhau của miền giá trị. Điều mà một hàm số phải thỏa mãn là yêu cầu rằng hai phần tử khác nhau của miền giá trị không thể cùng là ảnh của cùng một phần tử trong miền xác định
- Như vậy $f(x) = x^2$ là một hàm từ tập hợp số thực đến tập hợp số thực không âm (nó là toàn ánh nhưng không một-một), nhưng $f(x) = \pm \sqrt{x}$ không phải là một hàm số từ số thực không âm đến số thực

(g) Tính toán không chính xác các giá trị của hàm sàn (floor) và trần (ceiling), đặc biệt là đối với các giá trị âm

- Hàm sàn luôn làm tròn xuống, và hàm trần luôn làm tròn lên. Vì vậy, ví dụ, $\lfloor -3.2 \rfloor = -4$, và $\lceil -3.2 \rceil = -3$

12

Một số lỗi thường gặp

17

Một số lỗi thường gặp (tiếp)



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

Tập đếm được và không đếm được

(h) Sử dụng ký hiệu hàm sai, đặc biệt là với biến trong phương trình định nghĩa

- Cách viết đúng: Đầu vào nằm vào bên trong dấu ngoặc đơn theo sau tên hàm, và toàn bộ biểu thức sau đó biểu diễn đầu ra
- Ví dụ, nếu $f(x) = x^2$, thì việc viết $f(7) = 49$ là chính xác, và các cách viết không chính xác là $f = 49$ hoặc $f = 7$ hoặc $7 = 49$

(i) Nhầm lẫn ký hiệu hàm với phép nhân

- Mặc dù việc viết hai biểu thức toán học cạnh nhau, đặc biệt là với dấu ngoặc đơn, thường ám chỉ phép nhân, điều này chắc chắn không áp dụng khi đối tượng đầu tiên là một hàm số
- Ví dụ, nếu một hàm được định nghĩa bởi $f(x) = x + 10$, thì $f(7)$ có nghĩa là “ f của 7” (trong trường hợp này là 17) chứ không phải “ f nhân với 7” (điều này không có ý nghĩa)

(j) Cho rằng mọi hàm số đều là tuyến tính

13

Một số lỗi thường gặp

17

Một số lỗi thường gặp (tiếp)



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

Tập đếm được và không đếm được

- Luật phân phối cho các số cho biết rằng $t(x + y) = t(x) + t(y)$ nếu t , x , và y là các số và ký hiệu viết liền nhau biểu thị phép nhân. Ví dụ, $8(2 + 10) = 8 \cdot 2 + 8 \cdot 10$; cả hai vế đều bằng 96
- Tuy nhiên, nếu t là hàm bình phương và x và y vẫn là các số, thì $t(x + y) = t(x) + t(y)$ không đúng; nếu chúng ta lấy $x = 2$ và $y = 10$, thì vế trái là $(2 + 10)^2 = 12^2 = 144$, trong khi vế phải là $2^2 + 10^2 = 4 + 100 = 104$

(k) Áp dụng không chính xác trực giác về tập hữu hạn cho tập vô hạn

- Ví dụ, nếu một tập hữu hạn A có thể được đặt vào tương ứng một-một với một tập con thực sự của B , thì rõ ràng $|A| < |B|$. Tuy nhiên, điều này không đúng nếu A là tập vô hạn

14

Một số lỗi thường gặp

17

Một số lỗi thường gặp (tiếp)



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

Tập đếm được và không đếm được

- Chẳng hạn, xét A là tập các số tự nhiên chẵn và B là tập tất cả các số tự nhiên—khi đó A có thể được đặt vào tương ứng một-một với (thực tế là) một tập con thực sự của B , nhưng A cũng có thể được đặt vào tương ứng một-một với toàn bộ B (ghép cặp $2n$ với n cho $n = 0, 1, 2, \dots$), do đó thực tế $|A| = |B|$

(l) Không vẽ hình minh họa khi xử lý các quan hệ

- Đồ thị có hướng (digraph) của một quan hệ trên một tập hợp cung cấp một cách tuyệt vời để hình dung những gì đang diễn ra
- Lỗi phổ biến này có thể được tổng quát hóa: Không vẽ hình minh họa khi xử lý bất kỳ đối tượng toán học nào

(m) Quên xem xét các cặp (a, b) và (b, a) khi kiểm tra tính chất bắc cầu của một quan hệ

- Trong trường hợp này, người ta cần có (hoặc thêm vào) (a, a) và (b, b) nữa

15

Một số lỗi thường gặp

17

Một số lỗi thường gặp (tiếp)



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

Tập đếm được và không đếm được

(n) Không nhận ra rằng tính đối xứng hoặc tính bắc cầu thường đúng một cách tầm thường (vacuously)

- Ví dụ, quan hệ $\{(1, 2), (1, 3)\}$ thỏa mãn tính chất bắc cầu, bởi vì phát biểu “nếu (a, b) và (b, c) đều thuộc quan hệ thì (a, c) cũng thuộc quan hệ” đúng một cách tầm thường—tức là, không có cặp nào làm cho giả thiết của mệnh đề điều kiện này đúng

(o) Giả định không chính xác rằng mọi quan hệ đều có các tính chất mong muốn, như tính đối xứng hoặc tính bắc cầu

- Chắc chắn không phải mọi quan hệ đều là quan hệ tương đương
- Các phần tử có thể không so sánh được với nhau (ví dụ, không đúng rằng $\{1, 2\} \subseteq \{1, 3\}$ và cũng không đúng rằng $\{1, 3\} \subseteq \{1, 2\}$)
- Nếu tôi biết bạn và bạn biết Mary, điều đó không có nghĩa là tôi nhất định biết Mary

16

Một số lỗi thường gặp

17

Một số lỗi thường gặp (tiếp)



Các cấu trúc cơ bản

Hoàng Anh Đức

Tập hợp

Nghịch lý

Lực lượng của tập vô hạn

Định nghĩa

Tập đếm được và không đếm được

(p) Làm cho việc hiểu quan hệ tương đương trở nên phức tạp hơn mức cần thiết

- Trong hầu hết các trường hợp, bạn có thể nghĩ về một quan hệ tương đương như một quan hệ có dạng “hai phần tử có liên hệ với nhau khi và chỉ khi chúng có cùng một điều gì đó”
- Ví dụ, quan hệ tương đương trên tập hợp các số nguyên được cho bởi “ a có liên hệ với b khi và chỉ khi $a \equiv b \pmod{7}$ ” có thể được hiểu là “ a và b có liên hệ khi và chỉ khi chúng có cùng số dư khi chia cho 7”

(q) Suy nghĩ không đúng rằng một số toán tử phân phối qua các toán tử khác

- Ví dụ, không đúng rằng $\sqrt{a+b} = \sqrt{a} + \sqrt{b}$ hoặc $\sqrt{a^2 + b^2} = a + b$ hoặc $a^x + a^y = a^{x+y}$ hoặc $\log(a+b) = \log a + \log b$

17

Một số lỗi thường gặp

17

1. Logic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Đệ quy
4. Thuật toán I: Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp
5. Thuật toán II: Thuật toán tham lam, thuật toán đệ quy
6. Lý thuyết số cơ bản
7. Các phương pháp đếm
8. Lý thuyết đồ thị I: Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông
9. Lý thuyết đồ thị II: Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị
10. Lý thuyết đồ thị III: Cây
11. Đại số Boole

VNU-HUS MAT3500: Toán rời rạc

Quy nạp và Đệ quy

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn



Nội dung



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu
Quy nạp yếu
Quy nạp mạnh
Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm
Hàm định nghĩa bằng đệ quy
Tập hợp định nghĩa bằng đệ quy
Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi
Đoán nghiệm
Đa thức đặc trưng

Quy nạp toán học

Giới thiệu
Quy nạp yếu
Quy nạp mạnh
Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm
Hàm định nghĩa bằng đệ quy
Tập hợp định nghĩa bằng đệ quy
Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi
Đoán nghiệm
Đa thức đặc trưng

Quy nạp toán học

Giới thiệu



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

2

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

- *Quy nạp toán học (mathematical induction)* là một kỹ thuật chứng minh cực kỳ quan trọng
- Quy nạp toán học được sử dụng để *chứng minh các kết quả về những đối tượng rời rạc khác nhau*
- Ta sẽ giới thiệu một số dạng quy nạp toán học
 - *Quy nạp toán học yếu (Weak Mathematical Induction)*, hay còn gọi là *Nguyên lý Thứ nhất của Quy nạp toán học (The First Principle of Mathematical Induction)*
 - *Quy nạp toán học mạnh (Strong Mathematical Induction)* hay còn gọi là *Nguyên lý Thứ hai của Quy nạp toán học (The Second Principle of Mathematical Induction)*
- Tính đúng đắn của phương pháp quy nạp bắt nguồn từ

Tiên đề 1: Tính chất sắp thứ tự tốt

Mọi tập con khác rỗng của tập các số nguyên dương có một phần tử nhỏ nhất

Quy nạp toán học

Quy nạp yếu



Quy nạp và Đề quy

Hoàng Anh Đức

Nguyên lý quy nạp yếu

Để *chứng minh* $\forall n \in \mathbb{Z}^+ P(n)$, chúng ta thực hiện hai bước

- **Bước cơ sở (basis step):** Chỉ ra mệnh đề $P(1)$ đúng
- **Bước quy nạp (inductive step):** Chứng minh mệnh đề $P(k) \rightarrow P(k+1)$ đúng với mọi số nguyên dương k
 - Giả thiết $P(k)$ đúng, chứng minh $P(k+1)$ đúng
 - Giả thiết $P(k)$ đúng được gọi là *giả thiết quy nạp* (*inductive hypothesis hoặc induction hypothesis*)

Theo ngôn ngữ logic,

$$(P(1) \wedge \forall k \in \mathbb{Z}^+ (P(k) \rightarrow P(k+1))) \rightarrow \forall n \in \mathbb{Z}^+ P(n)$$

Quy nạp toán học

Giới thiệu

3

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng



■ Chứng minh $\forall n \in \mathbb{Z}^+ P(n)$ bằng phương pháp quy nạp

- (1) **Bước cơ sở:** Chứng minh $P(1)$ đúng
- (2) **Bước quy nạp:** Chứng minh $P(k) \rightarrow P(k+1)$ đúng với mọi $k \in \mathbb{Z}^+$. Theo định nghĩa của toán tử logic " $\rightarrow$ ", ta cần chứng minh rằng $P(k+1)$ **không thể sai khi $P(k)$ đúng**. Điều này có thể được thực hiện bằng cách **giả thiết là $P(k)$ đúng và chứng minh rằng với giả thiết đó $P(k+1)$ cũng đúng**

■ Chú ý rằng ở đây **ta không giả thiết $P(k)$ đúng với mọi $k \in \mathbb{Z}^+$**

■ Sau khi hoàn thành bước cơ sở và bước quy nạp:

- Từ bước cơ sở, ta biết rằng $P(1)$ đúng
- Từ bước quy nạp và $P(1)$ đúng, ta biết rằng $P(2)$ đúng
- Từ bước quy nạp và $P(2)$ đúng, ta biết rằng $P(3)$ đúng
- ...
- Từ bước quy nạp và $P(n-1)$ đúng, ta biết rằng $P(n)$ đúng (với bất kỳ số nguyên dương n)



Chứng minh (Quy nạp yếu là đúng).

- Giả sử $P(1)$ đúng và với mọi $k \in \mathbb{Z}^+$, $P(k) \rightarrow P(k+1)$ đúng. Ta chứng minh $\forall n \in \mathbb{Z}^+ P(n)$ bằng phản chứng
- Giả sử tồn tại $n \in \mathbb{Z}^+$ sao cho $P(n)$ sai. Do đó, tập $S = \{n \mid n \in \mathbb{Z}^+ \text{ và } P(n) \text{ sai}\} \subseteq \mathbb{Z}^+$ là tập khác rỗng.
- Theo Tiên đề 1, S có một phần tử nhỏ nhất m . Do $P(1)$ đúng, $m \neq 1$ và do đó $m > 1$, suy ra $m - 1 \in \mathbb{Z}^+$
- Do $m - 1 < m$, $m - 1 \notin S$, và do đó $P(m - 1)$ đúng
- Do $P(k) \rightarrow P(k + 1)$ đúng với mọi $k \in \mathbb{Z}^+$, ta có $P(m - 1) \rightarrow P(m)$ đúng.
- Kết hợp với $P(m - 1)$ đúng, ta có $P(m)$ đúng. Điều này mâu thuẫn với định nghĩa của m . Do đó $P(n)$ đúng với mọi số nguyên dương n



Quy nạp toán học

Chọn bước cơ sở



Quy nạp và Đề quy

Hoàng Anh Đức

Khi sử dụng quy nạp toán học, không nhất thiết cần bắt đầu với $P(1)$ ở bước cơ sở

Nguyên lý quy nạp yếu (tổng quát)

Để *chứng minh* $\forall n \geq b, P(n)$ với $n \in \mathbb{Z}$ và b là số nguyên cho trước, chúng ta thực hiện hai bước

- **Bước cơ sở (basis step):** Chỉ ra mệnh đề $P(b)$ đúng
- **Bước quy nạp (inductive step):** Chứng minh mệnh đề $P(k) \rightarrow P(k+1)$ đúng với mọi số nguyên $k \geq b$

Theo ngôn ngữ logic,

$$(P(b) \wedge \forall k \in \mathbb{Z}^{\geq b} (P(k) \rightarrow P(k+1))) \rightarrow \forall n \in \mathbb{Z}^{\geq b} P(n),$$

trong đó $\mathbb{Z}^{\geq b} = \{m \mid m \in \mathbb{Z} \text{ và } m \geq b\}$ (chú ý là $\mathbb{Z}^{\geq 1} = \mathbb{Z}^+$)

Quy nạp toán học

6

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Quy nạp toán học

Mẫu trình bày chứng minh quy nạp



- (1) Mô tả điều cần chứng minh dưới dạng “với mọi $n \geq b$, $P(n)$ ” với b là số nguyên cố định nào đó
 - “ $P(n)$ với mọi số nguyên dương n ” $\Rightarrow$ chọn $b = 1$
 - “ $P(n)$ với mọi số nguyên không âm n ” $\Rightarrow$ chọn $b = 0$
 - Với một số phát biểu, cần xác định giá trị phù hợp của b bằng cách kiểm tra giá trị chân lý của $P(n)$ với một số giá trị nhỏ của n
- (2) Viết cụm từ “Bước cơ sở.” Sau đó chỉ ra $P(b)$ là đúng. Hãy cẩn thận chọn đúng giá trị của b
- (3) Viết cụm từ “Bước quy nạp.” và phát biểu một cách rõ ràng giả thiết quy nạp dưới dạng “Giả sử rằng $P(k)$ đúng với một số nguyên cố định $k \geq b$ nào đó”
- (4) Phát biểu điều cần chứng minh với giả thiết $P(k)$ đúng, nghĩa là, phát biểu cụ thể $P(k+1)$
- (5) Chứng minh $P(k+1)$ đúng sử dụng giả thiết $P(k)$ đúng
- (6) Xác định rõ ràng phần kết của bước quy nạp, ví dụ như bằng cách viết “Bước quy nạp đến đây là hoàn tất.”
- (7) Sau khi hoàn thành bước cơ sở và bước quy nạp, phát biểu kết luận “Bằng phương pháp quy nạp, ta đã chứng minh $P(n)$ đúng với mọi số nguyên n thỏa mãn $n \geq b$.”

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

7

Quy nạp toán học

Chú ý



- Quy nạp toán học có thể được sử dụng để chứng minh một giả thuyết khi giả thuyết này đã được thành lập (và đúng). Tuy nhiên, quy nạp toán học **không cung cấp ý tưởng giải thích tại sao các định lý lại đúng**
- Việc **kiểm tra phát biểu cần chứng minh với một số giá trị nhỏ của n** trước khi đi vào chứng minh có thể rất hữu ích. Thông thường, các ví dụ nhỏ có thể giúp ta nhận ra các khía cạnh dễ nhầm lẫn của phát biểu hoặc nhận ra tại sao phát biểu lại đúng trong trường hợp tổng quát
- Thông thường, bước chứng minh $P(k+1)$ với giả thiết $P(k)$ là bước khó nhất trong toàn bộ chứng minh quy nạp. Hãy **chắc chắn rằng chứng minh của bạn đúng với mọi $k \geq b$** , nhất là với các giá trị nhỏ của k , thậm chí là cả $k = b$
- Nếu bạn **không sử dụng giả thiết $P(k)$ trong chứng minh $P(k+1)$** , thì có thể có điều gì đó sai, hoặc ít nhất chứng minh của bạn không thực sự là chứng minh quy nạp

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

8

63

Quy nạp toán học

Ví dụ



Ví dụ 1

Gọi $P(n)$ là vị từ $\sum_{i=1}^n i = \frac{n(n+1)}{2}$. Ta chứng minh $P(n)$ đúng với

mọi $n \in \mathbb{Z}^+$

■ **Bước cơ sở.** $P(1)$ đúng, do vế trái bằng 1 và vế phải bằng $1(1+1)/2 = 1$

■ **Bước quy nạp.** Giả sử $P(k)$ đúng với một số nguyên cố định

$k \geq 1$ nào đó, nghĩa là $\sum_{i=1}^k i = k(k+1)/2$. Ta chứng minh

$P(k+1)$ đúng, nghĩa là chứng minh $\sum_{i=1}^{k+1} i = (k+1)(k+2)/2$.

Thật vậy, ta có

$$\sum_{i=1}^{k+1} i = \sum_{i=1}^k i + (k+1) \quad \text{tách tổng}$$

$$= k(k+1)/2 + (k+1) \quad \text{giả thiết quy nạp}$$

$$= (k+1)(k+2)/2$$

Do đó, theo nguyên lý quy nạp yếu, $P(n)$ đúng với mọi $n \in \mathbb{Z}^+$

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

9 Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Quy nạp toán học

Ví dụ



Ví dụ 2

Gọi $P(n)$ là vị từ $\sum_{i=0}^n r^i = (r^{n+1} - 1)/(r - 1)$ với $r \neq 1$ là số thực bất kỳ cho trước. Ta chứng minh $P(n)$ đúng với mọi $n \in \mathbb{N}$

- **Bước cơ sở.** $P(0)$ đúng, do vế trái bằng 1 và vế phải bằng $(r^1 - 1)/(r - 1) = 1$
- **Bước quy nạp.** Giả sử $P(k)$ đúng với một số nguyên cố định

$k \geq 0$ nào đó, nghĩa là $\sum_{i=0}^k r^i = (r^{k+1} - 1)/(r - 1)$. Ta chứng

minh $P(k+1)$ đúng, nghĩa là chứng minh

$$\sum_{i=0}^{k+1} r^i = (r^{k+2} - 1)/(r - 1). \text{ Thật vậy, ta có}$$

$$\sum_{i=0}^{k+1} r^i = \sum_{i=0}^k r^i + r^{k+1}$$

tách tổng

$$= (r^{k+1} - 1)/(r - 1) + r^{k+1}$$

giả thiết quy nạp

$$= (r^{k+2} - 1)/(r - 1)$$

Do đó, theo nguyên lý quy nạp yếu, $P(n)$ đúng với mọi $n \in \mathbb{N}$

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Quy nạp toán học

Ví dụ



Ví dụ 3

Gọi $P(n)$ là vị từ

$$\sum_{i=0}^n (-1)^i = \begin{cases} 1 & \text{nếu } n \text{ chẵn} \\ 0 & \text{nếu } n \text{ lẻ} \end{cases}$$

Ta chứng minh $P(n)$ đúng với mọi số nguyên $n \geq 0$.

- **Bước cơ sở:** $P(0)$ đúng, do vế trái $\sum_{i=0}^0 (-1)^i = (-1)^0 = 1$ và vế phải bằng 1 (vì 0 là số chẵn)
- **Bước quy nạp:** Giả sử $P(k)$ đúng với một số nguyên cố định $k \geq 0$ nào đó, nghĩa là

$$\sum_{i=0}^k (-1)^i = \begin{cases} 1 & \text{nếu } k \text{ chẵn} \\ 0 & \text{nếu } k \text{ lẻ} \end{cases}$$

Ta chứng minh $P(k+1)$ đúng, nghĩa là chứng minh

$$\sum_{i=0}^{k+1} (-1)^i = \begin{cases} 1 & \text{nếu } k+1 \text{ chẵn} \\ 0 & \text{nếu } k+1 \text{ lẻ} \end{cases}$$

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

11

63

Quy nạp toán học

Ví dụ



Thật vậy, ta có

$$\sum_{i=0}^{k+1} (-1)^i = \sum_{i=0}^k (-1)^i + (-1)^{k+1}$$

$$= \begin{cases} 1 + (-1)^{k+1} & \text{nếu } k \text{ chẵn} \\ 0 + (-1)^{k+1} & \text{nếu } k \text{ lẻ} \end{cases}$$

$$= \begin{cases} 1 + (-1)^{k+1} & \text{nếu } k+1 \text{ lẻ} \\ 0 + (-1)^{k+1} & \text{nếu } k+1 \text{ chẵn} \end{cases}$$

$$= \begin{cases} 1 + (-1) & \text{nếu } k+1 \text{ lẻ} \\ 0 + 1 & \text{nếu } k+1 \text{ chẵn} \end{cases}$$

$$= \begin{cases} 0 & \text{nếu } k+1 \text{ lẻ} \\ 1 & \text{nếu } k+1 \text{ chẵn} \end{cases}$$

tách tổng

giả thiết quy nạp

k chẵn $\Leftrightarrow k+1$ lẻ

Do đó, theo nguyên lý quy nạp yếu, $P(n)$ đúng với mọi số nguyên $n \geq 0$

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

12

63



Bài tập 1

Cho $P(n)$ là phát biểu $1^2 + 2^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6}$

(a) Để chứng minh $P(n)$ đúng với mọi $n \in \mathbb{Z}^+$

(1) Ở bước cơ sở, ta cần chứng minh điều gì?

(2) Ở bước quy nạp, giả thiết quy nạp là gì? Ta cần chứng minh điều gì?

(b) Hãy chứng minh $P(n)$ đúng với mọi $n \in \mathbb{Z}^+$ bằng phương pháp quy nạp theo các bước bạn đã trả lời ở phần (a)

Bài tập 2

Chứng minh các mệnh đề sau bằng phương pháp quy nạp yếu

(a) $1^3 + 2^3 + \dots + n^3 = \left(\frac{n(n+1)}{2}\right)^2$ với mọi $n \in \mathbb{Z}^+$

(b) $1^2 + 3^2 + 5^2 + \dots + (2n+1)^2 = \frac{(n+1)(2n+1)(2n+3)}{3}$ với mọi $n \in \mathbb{N}$

(c) $3 + 3 \cdot 5 + 3 \cdot 5^2 + \dots + 3 \cdot 5^n = \frac{3(5^{n+1} - 1)}{4}$ với mọi $n \in \mathbb{N}$

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

13

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng



Bài tập 3

- (a) Chứng minh $3^n < n!$ với mọi số nguyên $n > 6$ bằng quy nạp yếu
- (b) Chứng minh rằng tồn tại số nguyên b sao cho $2n + 3 \leq 2^n$ với mọi $n \geq b$. (Gợi ý: Tìm b và chứng minh $2n + 3 \leq 2^n$ với mọi $n \geq b$ bằng phương pháp quy nạp yếu)
- (c) Chứng minh rằng $n^3 + 2n$ chia hết cho 3 với mọi số nguyên dương n bằng phương pháp quy nạp yếu

Quy nạp toán học

Bài tập



Hãy sử dụng phương pháp quy nạp yếu để giải các bài tập sau

Bài tập 4

Chứng minh với mọi số nguyên $n \geq 0$,

$$\sum_{i=0}^n 2^i = 2^{n+1} - 1$$

Bài tập 5

Chứng minh rằng với mọi số nguyên $n \geq 4$, ta có $2^n \geq n^2$

Bài tập 6

Tìm một số nguyên b thỏa mãn điều kiện với mọi $n \geq b$, $2^n \geq n^3$. Chứng minh rằng số b bạn tìm được thực sự thỏa mãn điều kiện đề ra

Bài tập 7

Chứng minh rằng với mọi số nguyên $n \geq 2$, các tập $A_1, A_2, \dots, A_n$ thỏa mãn $\overline{A_1 \cap A_2 \cap \dots \cap A_n} = \overline{A_1} \cup \overline{A_2} \cup \dots \cup \overline{A_n}$

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

15

63

Hãy sử dụng phương pháp quy nạp yếu để giải các bài tập sau

Bài tập 8

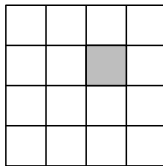
Chứng minh rằng với mọi $n \in \mathbb{Z}^+$, ta có thể phủ kín bàn cờ không hoàn chỉnh kích thước $2^n \times 2^n$ với một ô vuông bị loại bỏ bằng các khối hình chữ L như trong hình dưới đây sao cho không có hai khối nào chồng lên nhau¹



(a)



(b)



(c)

Hình: Phủ kín các bàn cờ, ví dụ như (b) hoặc (c), bằng các khối hình chữ L như ở (a)

¹ Bạn có thể thử phủ kín bàn cờ 8×8 ở <https://nstarr.people.amherst.edu/puzzle.html>

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

16

63



Hãy sử dụng phương pháp quy nạp yếu để giải các bài tập sau

Bài tập 9

Với các tập hợp $A_1, A_2, \dots, A_n$ và B , hãy chứng minh các phát biểu sau đúng với mọi số nguyên $n \geq 2$

$$(a) (A_1 \cap A_2 \cap \dots \cap A_n) \cup B = (A_1 \cup B) \cap (A_2 \cup B) \cap \dots \cap (A_n \cup B)$$

$$(b) (A_1 \cup A_2 \cup \dots \cup A_n) \cap B = (A_1 \cap B) \cup (A_2 \cap B) \cup \dots \cup (A_n \cap B)$$

$$(c) (A_1 - B) \cap (A_2 - B) \cap \dots \cap (A_n - B) = (A_1 \cap A_2 \cap \dots \cap A_n) - B$$

$$(d) (A_1 - B) \cup (A_2 - B) \cup \dots \cup (A_n - B) = (A_1 \cup A_2 \cup \dots \cup A_n) - B$$

Bài tập 10

Chứng minh rằng một tập có n phần tử có $n(n-1)/2$ tập con chứa chính xác hai phần tử, với mọi số nguyên $n \geq 2$

Bài tập 11

Chứng minh rằng $\neg(p_1 \vee p_2 \vee \dots \vee p_n) \equiv \neg p_1 \wedge \neg p_2 \wedge \dots \wedge \neg p_n$ với mọi $n \geq 2$, trong đó $p_1, \dots, p_n$ là các mệnh đề lôgic

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

17

63



Nguyên lý quy nạp mạnh

Để *chứng minh* $\forall n \in \mathbb{Z}^+ P(n)$, chúng ta thực hiện hai bước

- *Bước cơ sở (basis step)*: Chỉ ra mệnh đề $P(1)$ đúng
- *Bước quy nạp (inductive step)*: Chứng minh mệnh đề $(P(1) \wedge P(2) \wedge \cdots \wedge P(k)) \rightarrow P(k+1)$ đúng với mọi số nguyên dương k
 - Giả thiết $P(1) \wedge P(2) \wedge \cdots \wedge P(k)$ đúng, chứng minh $P(k+1)$ đúng
 - Giả thiết $P(1) \wedge P(2) \wedge \cdots \wedge P(k)$ đúng được gọi là *giả thiết quy nạp*

Theo ngôn ngữ logic,

$$(P(1) \wedge \forall k \in \mathbb{Z}^+ (\bigwedge_{j=1}^k P(j) \rightarrow P(k+1))) \rightarrow \forall n \in \mathbb{Z}^+ P(n),$$

trong đó $\bigwedge_{j=1}^k P(j) = P(1) \wedge P(2) \wedge \cdots \wedge P(k)$

Quy nạp toán học

Quy nạp mạnh



Tương tự như với quy nạp yếu, ở bước cơ sở ta không nhất thiết cần bắt đầu từ $P(1)$

Nguyên lý quy nạp mạnh (tổng quát)

Để *chứng minh* $\forall n \geq b P(n)$ với $n \in \mathbb{Z}$ và b là số nguyên cho trước, chúng ta thực hiện hai bước

- **Bước cơ sở (basis step):** Chỉ ra mệnh đề $P(b)$ đúng
- **Bước quy nạp (inductive step):** Chứng minh mệnh đề $(P(b) \wedge P(b+1) \wedge \dots \wedge P(k)) \rightarrow P(k+1)$ đúng với mọi số nguyên $k \geq b$

Theo ngôn ngữ logic,

$$(P(b) \wedge \forall k \in \mathbb{Z}^{\geq b} (\bigwedge_{j=b}^k P(j) \rightarrow P(k+1))) \rightarrow \forall n \in \mathbb{Z}^{\geq b} P(n)$$

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

19 Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Quy nạp toán học

Quy nạp mạnh



Thêm vào đó, thay vì chỉ chứng minh $P(b)$ đúng, ta có thể làm nhiều hơn ở bước cơ sở

Nguyên lý quy nạp mạnh (tổng quát)

Để **chứng minh** $\forall n \geq b P(n)$ với $n \in \mathbb{Z}$ và b là số nguyên cho trước, chúng ta thực hiện hai bước

- **Bước cơ sở (basis step):** Chỉ ra các mệnh đề $P(b)$, $P(b+1)$, $\dots$, $P(b+j)$ đúng, với j là một số nguyên dương cố định nào đó
- **Bước quy nạp (inductive step):** Chứng minh mệnh đề $(P(b) \wedge P(b+1) \wedge \dots \wedge P(k)) \rightarrow P(k+1)$ đúng với mọi số nguyên $k \geq b+j$

Theo ngôn ngữ logic,

$$\left(\bigwedge_{i=0}^j P(b+i) \wedge \forall k \in \mathbb{Z}^{\geq b+j} \left(\bigwedge_{i=b}^k P(i) \rightarrow P(k+1) \right) \right) \rightarrow \forall n \in \mathbb{Z}^{\geq b} P(n)$$

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

20

63



- Về mặt hình thức, quy nạp mạnh và quy nạp yếu **khác nhau ở giả thiết quy nạp**: ở quy nạp yếu ta chỉ giả thiết $P(k)$ đúng, còn ở quy nạp mạnh ta giả thiết tất cả các mệnh đề $P(1), P(2), \dots, P(k)$ đều đúng
- **Quy nạp mạnh và quy nạp yếu là tương đương về mặt lôgic**, nghĩa là, quy nạp yếu đúng khi và chỉ khi quy nạp mạnh cũng đúng
- Nếu bạn có thể trực tiếp chứng minh $P(k+1)$ với giả thiết $P(k)$ đúng, **không dùng quy nạp yếu**
- Ngược lại, nếu bạn có thể chứng minh $P(k+1)$ từ giả thiết $P(j)$ đúng với mọi $j \leq k$, nhưng không rõ làm sao để trực tiếp chứng minh $P(k) \rightarrow P(k+1)$, **không dùng quy nạp mạnh**



Ví dụ 4

Cho vị từ $P(n)$:

n viết được dưới dạng tích của một hoặc nhiều số nguyên tố

Ta chứng minh $\forall n \in \mathbb{Z}^{\geq 2} P(n)$ bằng quy nạp mạnh

- **Bước cơ sở.** $P(2)$ đúng, vì 2 có thể được biểu diễn dưới dạng tích của một số nguyên tố—chính nó
- **Bước quy nạp.** Giả sử $P(j)$ đúng với mọi số nguyên j thỏa mãn $2 \leq j \leq k$, trong đó $k \geq 2$ là số nguyên cố định nào đó. Ta chứng minh $P(k+1)$ đúng. Thật vậy,
 - Nếu $k+1$ là số nguyên tố, $P(k+1)$ đúng do $k+1$ có thể được biểu diễn dưới dạng tích của một số nguyên tố—chính nó
 - Nếu $k+1$ là hợp số, ta có thể biểu diễn $k+1 = ab$ với a, b là các số nguyên dương thỏa mãn $2 \leq a \leq b < k+1$. Theo giả thiết quy nạp, cả a và b đều có thể được biểu diễn dưới dạng tích của các số nguyên tố, và do đó $k+1$ cũng thế

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

22

63

Quy nạp toán học

Ví dụ



Ví dụ 5

Cho vị từ $P(n)$:

$$n = 4a + 5b \text{ với } a, b \in \mathbb{Z}$$

Ta chứng minh $\forall n \in \mathbb{Z}^{\geq 12} P(n)$ bằng quy nạp mạnh

- **Bước cơ sở.** Ta chứng minh $P(12), P(13), P(14), P(15)$ đều đúng. Thật vậy, $12 = 4 \cdot 3 + 5 \cdot 0$, $13 = 4 \cdot 2 + 5 \cdot 1$, $14 = 4 \cdot 1 + 5 \cdot 2$, và $15 = 4 \cdot 0 + 5 \cdot 3$
- **Bước quy nạp.** Giả sử với số nguyên cố định $k \geq 15$ bất kỳ, $P(m)$ đúng với mọi số nguyên m thỏa mãn $12 \leq m \leq k$. (Hay $P(12) \wedge \dots \wedge P(k)$ đúng với số nguyên cố định $k \geq 15$.) Ta chứng minh $P(k+1)$ đúng. Thật vậy, do $k \geq k-3 \geq 12$, theo giả thiết quy nạp, ta có $P(k-3)$ đúng, nghĩa là tồn tại $a, b \in \mathbb{Z}$ sao cho $k-3 = 4a + 5b$. Suy ra $k+1 = (k-3) + 4 = 4(a+1) + 5b$

Bài tập 12

Chứng minh ví dụ trên bằng quy nạp yếu

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

23

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Quy nạp toán học

Quy nạp mạnh



Sử dụng phương pháp quy nạp mạnh để giải các bài tập sau

Bài tập 13

Cho vị từ $P(n)$ = “tồn tại $a, b \in \mathbb{Z}$ thỏa mãn $n = 3a + 5b$ ”. Bài tập này mô tả cách chứng minh $P(n)$ đúng với mọi số nguyên $n \geq 8$

- (a) Để hoàn thành bước cơ sở, hãy chứng minh rằng các mệnh đề $P(8)$, $P(9)$, và $P(10)$ là đúng
- (b) Trong bước quy nạp, Giả thiết quy nạp là gì và bạn cần chứng minh điều gì?
- (c) Hãy hoàn thành bước quy nạp cho $n \geq 10$

Bài tập 14

Chứng minh rằng với mọi số nguyên $n \geq 12$, tồn tại các số nguyên không âm a và b sao cho $n = 3a + 7b$

Bài tập 15

Chứng minh rằng mọi số nguyên dương đều có thể được viết dưới dạng $2^b \cdot c$ trong đó b là một số nguyên không âm và c là một số lẻ

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

24

63

Quy nạp toán học

Một số chứng minh quy nạp sai



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

25 Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Ví dụ 6 ([Gunderson and Rosen 2010])

Cho $P(n)$ là $\sum_{i=1}^n i = \frac{(n + \frac{1}{2})^2}{2}$. Ta chứng minh $P(n)$ đúng với mọi

$n \in \mathbb{Z}^+$ bằng phương pháp quy nạp yếu

■ **Bước cơ sở.** $P(1)$ đúng do $\frac{(1 + \frac{1}{2})^2}{2} = 1$

■ **Bước quy nạp.** Giả sử $P(k)$ đúng với một số nguyên cố định $k \geq 1$ nào đó. Ta chứng minh $P(k+1)$ đúng. Thật vậy

$$\sum_{i=1}^{k+1} i = \left(\sum_{i=1}^k i \right) + (k+1) \quad \text{tách tổng}$$

$$= \frac{(k + \frac{1}{2})^2}{2} + (k+1) \quad \text{giả thiết quy nạp}$$

$$= \frac{(k + \frac{1}{2})^2 + 2(k + \frac{1}{2}) + 1}{2}$$

$$= \frac{(k+1 + \frac{1}{2})^2}{2}$$

Câu hỏi

Chứng minh này sai ở đâu?

Do đó, theo nguyên lý quy nạp yếu, $P(n)$ đúng với mọi $n \in \mathbb{Z}^+$



Ví dụ 7 ([Gunderson and Rosen 2010])

Cho vị từ $P(n)$

Nếu $n = \max\{a, b\}$ với $a, b \in \mathbb{Z}^+$ thì $a = b$

Ta chứng minh $P(n)$ đúng với mọi $n \in \mathbb{Z}^+$ bằng quy nạp yếu

■ **Bước cơ sở.** Với $n = 1$, giả sử $1 = \max\{a, b\}$ với $a, b \in \mathbb{Z}^+$. Do đó $a = b = 1$. Suy ra $P(1)$ đúng

■ **Bước quy nạp.** Giả sử $P(k)$ đúng với một số nguyên cố định $k \geq 1$ nào đó, nghĩa là nếu $k = \max\{a, b\}$ với $a, b \in \mathbb{Z}^+$ thì $a = b$

Ta chứng minh $P(k+1)$ đúng, nghĩa là chứng minh nếu $k+1 = \max\{c, d\}$ với $c, d \in \mathbb{Z}^+$ thì $c = d$. Thật vậy, giả sử hai số $c, d \in \mathbb{Z}^+$ thỏa mãn $k+1 = \max\{c, d\}$. Do đó, $\max\{c-1, d-1\} = k$. Theo giả thiết quy nạp, $c-1 = d-1$, và do đó $c = d$. Theo nguyên lý quy nạp yếu, $P(n)$ đúng với mọi $n \in \mathbb{Z}^+$

Câu hỏi

Chứng minh này sai ở đâu?

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

26

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Quy nạp toán học

Một số chứng minh quy nạp sai



Ví dụ 8 (Tất cả ngựa đều cùng màu)

$P(n) :=$ “*Bất kỳ n con ngựa nào đều có cùng màu sắc*”

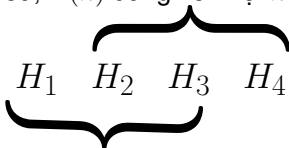
Thật vậy, ta chứng minh $\forall n \in \mathbb{Z}^+ P(n)$ bằng phương pháp quy nạp yếu

- **Bước cơ sở.** $P(1)$ hiển nhiên đúng.
- **Bước quy nạp.** Giả sử $P(k)$ đúng với số nguyên $k \geq 1$ cố định nào đó. Ta chứng minh $P(k+1)$ cũng đúng. Giả sử có $k+1$ con ngựa $H_1, H_2, \dots, H_k, H_{k+1}$. Theo giả thiết quy nạp, $H_1, H_2, \dots, H_k$ đều có cùng màu sắc. Cũng theo giả thiết quy nạp, $H_2, \dots, H_k, H_{k+1}$ đều có cùng màu sắc. Do đó, $H_1, H_2, \dots, H_k, H_{k+1}$ đều có cùng màu sắc

Do đó, theo nguyên lý quy nạp yếu, $P(n)$ đúng với mọi $n \in \mathbb{Z}^+$

Câu hỏi

Chứng minh này sai ở đâu?



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

27

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng



Ví dụ 9

Câu hỏi

Chứng minh này sai ở đâu?

$$P(n) := "5n = 0"$$

Ta chứng minh $\forall n \in \mathbb{N} P(n)$ bằng quy nạp mạnh

- **Bước cơ sở:** $P(0)$ đúng, do $5 \cdot 0 = 0$
- **Bước quy nạp:** Giả sử với số nguyên $k \geq 0$ cố định nào đó, $5j = 0$ với mọi số nguyên không âm j thỏa mãn $0 \leq j \leq k$. Ta chứng minh $5(k+1) = 0$. Thật vậy, ta có thể viết $k+1 = i+j$, với $i, j \in \mathbb{N}$ và i, j nhỏ hơn $k+1$. Theo giả thiết quy nạp, $5(k+1) = 5(i+j) = 5i + 5j = 0 + 0 = 0$

Do đó, theo nguyên lý quy nạp mạnh, $P(n)$ đúng với mọi $n \in \mathbb{N}$

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

28

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Đệ quy

Định nghĩa và một số khái niệm



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

29

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

- Trong **quy nạp**, ta chứng minh mọi phần tử của một tập vô hạn thỏa mãn vị từ P nào đó bằng cách
 - chứng minh tính đúng đắn của P cho các phần tử lớn hơn trong tập hợp dựa vào tính đúng đắn của các phần tử nhỏ hơn
- Trong các **định nghĩa đệ quy (recursive definition)**, tương tự, ta định nghĩa một cấu trúc (hàm, vị từ, tập hợp, hay một cấu trúc nào đó phức tạp hơn) trên một miền vô hạn nào đó (miền xác định) bằng cách
 - định nghĩa cấu trúc của các phần tử lớn hơn dựa vào cấu trúc của các phần tử nhỏ hơn

Đệ quy

Hàm định nghĩa bằng đệ quy



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

■ Hàm định nghĩa bằng đệ quy (recursive function)

$f : \mathbb{N} \rightarrow A$ với tập A bất kỳ

■ **Bước cơ sở (basis step):** Định nghĩa một số giá trị ban đầu $f(0), f(1), \dots, f(b)$ với số nguyên cố định $b \geq 0$ nào đó

■ **Bước đệ quy (recursive step):** Định nghĩa một quy luật để tìm giá trị của $f(n)$ từ các giá trị $f(n-1), f(n-2), \dots, f(n-b), f(n-b-1)$ với mọi $n > b$

■ Các tính chất của hàm định nghĩa bằng đệ quy thường được chứng minh bằng quy nạp

Ví dụ 10

Định nghĩa một dãy bằng hệ thức truy hồi

■ Dãy Fibonacci $\{f_n\}$

■ **Bước cơ sở:** $f_0 = 0, f_1 = 1$

■ **Bước đệ quy:** $f_n = f_{n-1} + f_{n-2} \ (n \geq 2)$

■ Dãy giai thừa $\{g_n\}$

■ **Bước cơ sở:** $g_0 = 1$

■ **Bước đệ quy:** $g_n = ng_{n-1} \ (n \geq 1)$

30

63

Đệ quy

Hàm định nghĩa bằng đệ quy



Ví dụ 11 (Chứng minh tính chất của hàm sử dụng định nghĩa đệ quy)

Cho dãy Fibonacci $\{f_n\}$. Ta chứng minh $f_n < 2^n$ với mọi $n \in \mathbb{N}$ bằng quy nạp mạnh

- **Bước cơ sở:** $f_0 = 0 < 2^0 = 1$ và $f_1 = 1 < 2^1 = 2$ (sử dụng các định nghĩa ở bước cơ sở của định nghĩa đệ quy)
- **Bước quy nạp:** Giả sử với số nguyên $k \geq 0$ cố định nào đó, $f_i < 2^i$ với mọi i thỏa mãn $1 \leq i \leq k$. Ta chứng minh $f_{k+1} < 2^{k+1}$. Thật vậy,

$$\begin{aligned}f_{k+1} &= f_k + f_{k-1} \\&< 2^k + 2^{k-1} \\&< 2^k + 2^k = 2^{k+1}\end{aligned}$$

định nghĩa đệ quy của $\{f_n\}$
giả thiết quy nạp

Do đó, theo nguyên lý quy nạp mạnh, $f_n < 2^n$ với mọi $n \in \mathbb{N}$

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

31

63

Đệ quy

Hàm định nghĩa bằng đệ quy



Ví dụ 12 (Chứng minh tính chất của hàm sử dụng định nghĩa đệ quy)

Cho dãy Fibonacci $\{f_n\}$. Ta chứng minh $f_n > \alpha^{n-2}$ với mọi số tự nhiên $n \geq 3$ và $\alpha = \frac{1 + \sqrt{5}}{2} \approx 1.61803$ bằng quy nạp mạnh

■ **Bước cơ sở:** Với $n = 3$, ta có $f_3 = 2 > \alpha^{3-2} = \alpha$. Với $n = 4$, ta có $f_4 = 3 > \alpha^{4-2} = \alpha^2 = \left(\frac{1 + \sqrt{5}}{2}\right)^2 = \frac{3 + \sqrt{5}}{2} = \alpha + 1 \approx 2.61803$

■ **Bước quy nạp:** Giả sử với số nguyên $k \geq 3$ cố định nào đó, $3 \leq i \leq k$ với mọi i thỏa mãn $f_i > \alpha^{i-2}$. Ta chứng minh $f_{k+1} > \alpha^{k-1}$. Thật vậy,

$$f_{k+1} = f_k + f_{k-1}$$

$$> \alpha^{k-2} + \alpha^{k-3}$$

$$= \alpha^{k-3}(\alpha + 1)$$

$$= \alpha^{k-3}\alpha^2 = \alpha^{k-1}$$

định nghĩa đệ quy của $\{f_n\}$

giả thiết quy nạp

do $\alpha^2 = \alpha + 1$

Do đó, theo nguyên lý quy nạp mạnh, $f_n > \alpha^{n-2}$ với mọi số tự nhiên $n \geq 3$

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

32

63

Đệ quy

Hàm định nghĩa bằng đệ quy



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Bài tập 16

Cho các hàm sau. Hãy xây dựng định nghĩa đệ quy cho mỗi hàm (ghi rõ bước cơ sở và bước đệ quy).

- (a) Hàm $F(n)$ là tổng của n số nguyên dương đầu tiên.
- (b) Hàm $S_m(n)$ là tổng của số nguyên m và số nguyên không âm n .
- (c) Hàm $P_m(n)$ là tích của số nguyên m và số nguyên không âm n .

Bài tập 17

Cho dãy Fibonacci $\{f_n\}$. Chứng minh rằng

- (a) $f_1^2 + f_2^2 + \cdots + f_n^2 = f_n f_{n+1}$ với mọi số nguyên dương n
- (b) $f_1 + f_3 + \cdots + f_{2n-1} = f_{2n}$ với mọi số nguyên dương n

33

63

Đệ quy

Tập hợp định nghĩa bằng đệ quy



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

34

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

- **Tập hợp định nghĩa bằng đệ quy (recursive set)**
 - **Bước cơ sở (basis step):** Định nghĩa một tập con các phần tử ban đầu
 - **Bước đệ quy (recursive step):** Định nghĩa một quy luật để tìm phần tử mới trong tập từ các phần tử đã biết là thuộc tập đó
- Thông thường, với các tập định nghĩa bằng đệ quy, **quy tắc ngoại trừ (exclusion rule)** sau luôn được áp dụng: tập hợp cần định nghĩa chỉ chứa các phần tử liệt kê ở bước cơ sở và các phần tử thu được bằng cách áp dụng quy tắc ở bước đệ quy.

Đệ quy

Tập hợp định nghĩa bằng đệ quy



Ví dụ 13

- Tập S các số nguyên dương chia hết cho 3
 - **Bước cơ sở:** $3 \in S$
 - **Bước đệ quy:** Nếu $x \in S$ và $y \in S$ thì $x + y \in S$
 - Đầu tiên, $3 \in S$, sau đó là $3 + 3 = 6$, $3 + 6 = 9$, v.v...
- Tập số tự nhiên $\mathbb{N}$
 - **Bước cơ sở:** $0 \in \mathbb{N}$
 - **Bước đệ quy:** Nếu $n \in \mathbb{N}$ thì $n + 1 \in \mathbb{N}$
- Tập các chuỗi ký tự Σ^* sinh bởi bảng chữ cái Σ
 - **Bước cơ sở:** $\lambda \in \Sigma^*$ (λ là chuỗi rỗng không chứa bất kỳ ký tự nào)
 - **Bước đệ quy:** Nếu $w \in \Sigma^*$ và $x \in \Sigma$ thì $wx \in \Sigma^*$
 - Ví dụ nếu $\Sigma = \{0, 1\}$ thì
 - $\lambda \in \Sigma^*$ (bước cơ sở)
 - $\{0, 1\} \subseteq \Sigma^*$ (lần đầu áp dụng bước quy nạp)
 - $\{00, 01, 10, 11\} \subseteq \Sigma^*$ (lần thứ hai áp dụng bước quy nạp)
 - v.v...
 - Do đó Σ^* là tập tất cả các chuỗi nhị phân

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

35

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Đệ quy

Tập hợp định nghĩa bằng đệ quy



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

36

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Ví dụ 14

- Tập *các công thức được tạo đúng quy tắc (well-formed formulae)* trong logic mệnh đề
 - **Bước cơ sở:** $\mathbf{T}$, $\mathbf{F}$, và mệnh đề nguyên tử s là các công thức được tạo đúng quy tắc
 - **Bước đệ quy:** Nếu A và B là các công thức được tạo đúng quy tắc, thì $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, và $(A \leftrightarrow B)$ cũng thế
 - Ví dụ, với các mệnh đề nguyên tử p, q , $((p \vee q) \rightarrow (q \wedge \mathbf{F}))$ và $(p \wedge q)$ là các công thức được tạo đúng quy tắc, còn $\wedge pq$, $pq\wedge$, và $p \wedge q$ thì không phải (**Tại sao?**)

Đệ quy

Tập hợp định nghĩa bằng đệ quy



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

37 Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Đẩy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Bài tập 18

Hãy tìm một định nghĩa đệ quy của

- (a) Dãy $\{a_n\}$ với $a_n = 4n - 2$ và $n = 1, 2, \dots$
- (b) Dãy $\{b_n\}$ với $b_n = n(n + 1)$ và $n = 1, 2, \dots$
- (c) Tập hợp các số nguyên dương lẻ
- (d) Tập hợp các số nguyên dương là lũy thừa của 3
- (e) Tập hợp các số nguyên dương chia hết cho 5
- (f) Tập hợp các số nguyên dương không chia hết cho 5

Đệ quy

Quy nạp theo cấu trúc



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

38 Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Để chứng minh một tính chất P của các phần tử của một tập hợp định nghĩa theo đệ quy, ta sử dụng **quy nạp theo cấu trúc** (*structural induction*)

Nguyên lý quy nạp theo cấu trúc

- **Bước cơ sở:** Chứng minh rằng mọi phần tử định nghĩa trong bước cơ sở của định nghĩa đệ quy đều thỏa mãn P
- **Bước quy nạp:** Chứng minh rằng nếu các phần tử được sử dụng để xây dựng phần tử mới của tập hợp trong bước đệ quy đều thỏa mãn P , thì phần tử mới cũng thỏa mãn P

Đệ quy

Quy nạp theo cấu trúc



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

39

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Ví dụ 15

Cho S là tập định nghĩa theo đệ quy như sau:

■ **Bước cơ sở:** $3 \in S$

■ **Bước đệ quy:** Nếu $x \in S$ và $y \in S$ thì $x + y \in S$

Ta chứng minh bằng quy nạp theo cấu trúc rằng *mọi phần tử của S đều chia hết cho 3*

■ **Bước cơ sở:** 3 chia hết cho 3

■ **Bước quy nạp:** Giả sử với $x \in S$ và $y \in S$, cả x và y đều chia hết cho 3. Ta chứng minh $n = x + y$ cũng chia hết cho 3. Thật vậy, do x chia hết cho 3, ta có $x = 3k$ với số nguyên k nào đó. Tương tự, $y = 3j$ với số nguyên j nào đó. Suy ra $n = x + y = 3(k + j)$, và do đó n cũng chia hết cho 3

Do đó, theo nguyên lý quy nạp theo cấu trúc, mọi phần tử của S đều chia hết cho 3

Đệ quy

Quy nạp theo cấu trúc



Ví dụ 16

Ta chứng minh bằng quy nạp theo cấu trúc rằng *mọi công thức được tạo đúng quy tắc trong logic mệnh đề có số dấu ngoặc đơn trái "(" bằng số dấu ngoặc đơn phải ")"*

- **Bước cơ sở:** Các mệnh đề T , F , và mọi mệnh đề nguyên tử s đều không có các dấu ngoặc đơn trái và phải
- **Bước quy nạp:** Với công thức A , gọi l_A và r_A lần lượt là số ngoặc đơn trái và ngoặc đơn phải của A . Giả sử với các công thức A, B , $l_A = r_A$ và $l_B = r_B$. Ta chứng minh rằng điều này cũng đúng với các công thức $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, và $(A \leftrightarrow B)$. Thật vậy, công thức đầu tiên có $l_A + 1$ ngoặc trái và $r_A + 1$ ngoặc phải, và các công thức sau đó có $l_A + l_B + 1$ ngoặc trái và $r_A + r_B + 1$ ngoặc phải

Do đó, theo nguyên lý quy nạp theo cấu trúc, mọi công thức được tạo đúng quy tắc trong logic mệnh đề có số dấu ngoặc đơn trái bằng số dấu ngoặc đơn phải

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

40

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Đệ quy

Quy nạp theo cấu trúc



Bài tập 19

Chứng minh rằng tập S định nghĩa bởi $1 \in S$ và $s + t \in S$ nếu $s \in S$ và $t \in S$ là tập các số nguyên dương $\mathbb{Z}^+$. (**Gợi ý:** Chứng minh $S \subseteq \mathbb{Z}^+$ và $\mathbb{Z}^+ \subseteq S$)

Bài tập 20

Cho S là tập các cặp sắp thứ tự các số nguyên được định nghĩa bằng đệ quy như sau

- **Bước cơ sở:** $(0, 0) \in S$
 - **Bước đệ quy:** Nếu $(a, b) \in S$, thì $(a + 2, b + 3) \in S$ và $(a + 3, b + 2) \in S$
- (a) Sử dụng quy nạp mạnh với số lần áp dụng bước đệ quy trong định nghĩa của S ở trên, hãy chứng minh $a + b$ chia hết cho 5 với mọi $(a, b) \in S$
- (b) Sử dụng quy nạp theo cấu trúc để chứng minh $a + b$ chia hết cho 5 với mọi $(a, b) \in S$

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

41

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

- Một **hệ thức truy hồi (recurrence relation)** cho dãy $\{a_n\}$ là một **phương trình biểu diễn a_n thông qua một hoặc nhiều số hạng trước đó của dãy** với mọi số nguyên n thỏa mãn $n \geq n_0$ với n_0 là một số nguyên không âm.
 - Với dãy $\{a_n\} = 0, 1, 4, 9, 16 \dots (n \geq 0)$, $a_n = a_{n-1} + 2n - 1$ với $n \geq 1$ là một hệ thức truy hồi cho $\{a_n\}$ (ở đây $n_0 = 1$)
- Để **định nghĩa một dãy $\{a_n\}$ thông qua hệ thức truy hồi**, ta cần thêm **các điều kiện ban đầu (initial conditions)** bằng cách định nghĩa các phần tử trước a_{n_0} trong dãy
 - Để **định nghĩa $\{a_n\}$ qua hệ thức $a_n = a_{n-1} + 2n - 1$ ($n \geq 1$)**, ta cần thêm **điều kiện ban đầu $a_0 = 0$**
- Một dãy được gọi là một **ng nghiệm (solution)** của một hệ thức truy hồi nếu **các số hạng của dãy thỏa mãn hệ thức đó**
- **Giải hệ thức truy hồi với các điều kiện ban đầu** nghĩa là **tìm một công thức tường minh cho các số hạng của dãy**
 - Một công thức tường minh cho dãy $\{a_n\}$ định nghĩa bởi $a_n = a_{n-1} + 2n - 1$ với $n \geq 1$ và điều kiện ban đầu $a_0 = 0$ là $a_n = n^2$ ($n \geq 0$)

42

63

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Ví dụ 17

- **Dãy $\{b_n\}$ thỏa mãn hệ thức truy hồi $b_n = -b_{n-1}$ với $n \geq 1$ và điều kiện ban đầu $b_0 = 1$**

- $\{b_n\} = 1, -1, 1, -1, \dots$

- **Dãy $\{s_n\}$ thỏa mãn hệ thức truy hồi $s_n = s_{n-1} - s_{n-2}$ với $n \geq 2$ và điều kiện ban đầu $s_0 = 3$ và $s_1 = 5$**

- $\{s_n\} = 3, 5, 2, -3, -5, \dots$

- ***Dãy Fibonacci (Fibonacci sequence)* $\{f_n\}$ ($n \geq 0$) được định nghĩa bởi điều kiện ban đầu $f_0 = 0, f_1 = 1$ và hệ thức truy hồi $f_n = f_{n-1} + f_{n-2}$ với mọi số nguyên $n \geq 2$**

- $\{f_n\} = 0, 1, 1, 2, 3, 5, \dots$

- ***Dãy giai thừa (factorial sequence)* $\{g_n\}$ được định nghĩa bởi điều kiện ban đầu $g_0 = 1$ và hệ thức truy hồi $g_n = ng_{n-1}$ với mọi số nguyên $n \geq 1$**

- $\{g_n\} = 1, 1, 2, 6, 24, \dots$

43

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy
nap sai

Đề quy

Định nghĩa và một số khái
niệm

Hàm định nghĩa bằng đệ
quy

Tập hợp định nghĩa bằng
đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

44

Dãy cho bởi hệ thức truy
hồi

Đoán nghiệm

Đa thức đặc trưng

■ Một số phương pháp giải hệ thức truy hồi

- (1) Đoán nghiệm (và chứng minh bằng phương pháp quy nạp)
- (2) Sử dụng đa thức đặc trưng
- (3) Sử dụng hàm sinh

Giải hệ thức truy hồi

Đoán nghiệm



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

45

Đoán nghiệm

Đa thức đặc trưng

- **Đoán nghiệm (Guessing and verifying)** là phương pháp giải hệ thức truy hồi bằng cách:
 - (1) Tính một số giá trị đầu tiên của dãy từ hệ thức truy hồi
 - (2) Quan sát các giá trị này để **đoán** một công thức tường minh
 - (3) Chứng minh công thức đoán được là đúng bằng **phương pháp quy nạp toán học**
- Kỹ thuật đoán nghiệm thường dựa vào:
 - Thay lập đi lập lại hệ thức truy hồi cho các số hạng đứng trước
 - Nhận ra quy luật xuất hiện trong quá trình thay thế
 - Tổng quát hóa quy luật này thành một công thức

Giải hệ thức truy hồi

Đoán nghiệm



Ví dụ 18

Giải hệ thức truy hồi $d_n = d_{n-1} + 4$ ($n \geq 1$) với điều kiện ban đầu $d_0 = -1$

Hướng suy luận

- (1) Từ hệ thức truy hồi, ta cũng có $d_{n-1} = d_{n-2} + 4$
- (2) Thay (2) vào hệ thức ban đầu
$$d_n = (d_{n-2} + 4) + 4 = d_{n-2} + 2 \cdot 4$$
- (3) Từ hệ thức truy hồi, ta cũng có $d_{n-2} = d_{n-3} + 4$
- (4) Thay (4) vào (3), ta thu được $d_n = d_{n-3} + 3 \cdot 4$
- (5) Lặp lại quá trình trên, ta “đoán” $d_n = d_{n-r} + r \cdot 4$
- (6) Để có một công thức tường minh cho d_n , ta cần $n - r = 0$, tức là $r = n$. Khi đó d_n được biểu diễn qua d_0 đã cho trước và $n = r$.
- (7) Tóm lại, ta có $d_n = -1 + 4n$

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

46

Đoán nghiệm

Đa thức đặc trưng

Giải hệ thức truy hồi

Đoán nghiệm



Ví dụ 19

Giải hệ thức truy hồi $a_n = a_{n-1} + 2n - 1$ ($n \geq 1$) với điều kiện ban đầu $a_0 = 0$

(1) Từ hệ thức truy hồi, ta có $a_{n-1} = a_{n-2} + 2(n-1) - 1$

(2) Thay vào hệ thức ban đầu,

$$a_n = (a_{n-2} + 2(n-1) - 1) + 2n - 1 = a_{n-2} + 4n - 4$$

(3) Từ hệ thức truy hồi, ta có $a_{n-2} = a_{n-3} + 2(n-2) - 1$

(4) Thay vào (2),

$$a_n = (a_{n-3} + 2(n-2) - 1) + 4n - 4 = a_{n-3} + 6n - 9$$

(5) Từ hệ thức truy hồi, ta có $a_{n-3} = a_{n-4} + 2(n-3) - 1$

(6) Thay vào (4),

$$a_n = (a_{n-4} + 2(n-3) - 1) + 6n - 9 = a_{n-4} + 8n - 16$$

(7) Lặp lại quá trình trên, ta “đoán” $a_n =$

(8) Để có một công thức tường minh cho a_n , ta cần $n - r = 0$, tức là $r = n$. Khi đó a_n được biểu diễn qua a_0 đã cho trước và $n = r$

(9) Tóm lại $a_n =$

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

47

Đoán nghiệm

Đa thức đặc trưng

Giải hệ thức truy hồi

Đoán nghiệm



Bài tập 21

Giải các hệ thức truy hồi với điều kiện ban đầu sau bằng phương pháp đoán nghiệm và chứng minh công thức bạn đoán được là đúng bằng quy nạp

(a) $a_n = -a_{n-1}, a_0 = 5$

(b) $a_n = a_{n-1} + 3, a_0 = 1$

(c) $a_n = a_{n-1} - n, a_0 = 4$

(d) $a_n = 5a_{n-1}, a_0 = 1$

(e) $a_n = 2a_{n-1} - 3, a_0 = -1$

(f) $a_n = (n+1)a_{n-1}, a_0 = 2$

(g) $a_n = 2na_{n-1}, a_0 = 3$

(h) $a_n = -a_{n-1} + n - 1, a_0 = 7$

Bài tập 22

Giả thiết dân số thế giới năm 2023 là 8 tỷ người và tăng theo tỷ lệ 0.8%/năm. Với $n \geq 1$,

(a) Xây dựng hệ thức truy hồi để tính dân số thế giới n năm sau 2023

(b) Tìm công thức tường minh để tính dân số thế giới n năm sau 2023

(c) Dân số thế giới năm 2057 sẽ là bao nhiêu?

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

48

Đoán nghiệm

Đa thức đặc trưng

63

Giải hệ thức truy hồi

Đa thức đặc trưng



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

- **Hệ thức truy hồi tuyến tính thuần nhất bậc k với hệ số hằng (Linear homogeneous recurrence relation of degree k with constant coefficients)** là hệ thức có dạng

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k},$$

trong đó $c_1, c_2, \dots, c_k$ là các số thực, $c_k \neq 0$, và $n \geq k$

- Một dãy $\{a_n\}$ ($n \geq 0$) thỏa mãn hệ thức truy hồi trên được **xác định một cách duy nhất** bởi **hệ thức này** và **k điều kiện ban đầu** $a_0 = C_0, a_1 = C_1, \dots, a_{k-1} = C_{k-1}$

- Hệ thức truy hồi $f_n = f_{n-1} + f_{n-2}$ là một hệ thức truy hồi tuyến tính thuần nhất bậc hai với hệ số hằng. Dãy Fibonacci $\{f_n\}$ được xác định bởi hệ thức trên và điều kiện ban đầu $f_0 = 0, f_1 = 1$
- Các hệ thức $g_n = n g_{n-1}$, $g_n = g_{n-1} + g_{n-2}^2$ không là hệ thức truy hồi tuyến tính thuần nhất với hệ số hằng

49

63

Giải hệ thức truy hồi

Đa thức đặc trưng



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

- Hệ thức $a_n = c_1 a_{n-1} + \dots + c_k a_{n-k}$ ($c_k \neq 0$) có hai tính chất quan trọng

- Nghiệm của hệ thức có dạng $a_n = r^n$ với $r \neq 0$ là một hằng số nào đó. Từ đó, $r^n = c_1 r^{n-1} + \dots + c_k r^{n-k}$. Chia hai vế cho r^{n-k} và chuyển vế, ta có

$$r^k - c_1 r^{k-1} - c_2 r^{k-2} - \dots - c_{k-1} r - c_k = 0.$$

Đa thức cuối cùng gọi là **đa thức đặc trưng (characteristic polynomial)** của hệ thức, và nghiệm của nó gọi là **nghiệm đặc trưng (characteristic root)** của hệ thức

- Nếu s_n và t_n thỏa mãn hệ thức truy hồi, thì **mọi tổ hợp tuyến tính của s_n và t_n** , nghĩa là mọi biểu thức có dạng $b_1 s_n + b_2 t_n$ với b_1, b_2 là các số thực nào đó, cũng thỏa mãn hệ thức truy hồi

50

Giải hệ thức truy hồi

Đa thức đặc trưng



Quy nạp và Đệ quy

Hoàng Anh Đức

Định lý 2

Cho các số thực $c_1, c_2, \dots, c_k$. Giả sử $r^k - c_1 r^{k-1} - \dots - c_{k-1} r - c_k$ có t nghiệm phân biệt $r_1, r_2, \dots, r_t$ với các bội tương ứng $m_1, m_2, \dots, m_t$ thỏa mãn $m_i \geq 1$ với $1 \leq i \leq t$ và $m_1 + \dots + m_t = k$ (nghĩa là, có m_i nghiệm có giá trị r_i). Dãy $\{a_n\}$ là nghiệm của hệ thức truy hồi $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$ ($n \geq k$) với điều kiện ban đầu $a_0 = C_0, a_1 = C_1, \dots, a_{k-1} = C_{k-1}$ khi và chỉ khi

$$\begin{aligned} a_n = & (\alpha_{1,0} + \alpha_{1,1}n + \dots + \alpha_{1,m_1-1}n^{m_1-1})r_1^n \\ & + (\alpha_{2,0} + \alpha_{2,1}n + \dots + \alpha_{2,m_2-1}n^{m_2-1})r_2^n \\ & + \dots + (\alpha_{t,0} + \alpha_{t,1}n + \dots + \alpha_{t,m_t-1}n^{m_t-1})r_t^n \end{aligned}$$

với $n \geq 0$, trong đó $\alpha_{i,j}$ là các hằng số với $1 \leq i \leq t$ và $0 \leq j \leq m_i - 1$

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

51

Giải hệ thức truy hồi

Đa thức đặc trưng



Các bước giải hệ thức truy hồi tuyến tính thuần nhất bậc k với hệ số hằng

- (1) Xác định đa thức đặc trưng: $r^k - c_1 r^{k-1} - \dots - c_{k-1} r - c_k$
- (2) Tìm các nghiệm $r_1, r_2, \dots, r_t$ của đa thức đặc trưng và bội tương ứng $m_1, m_2, \dots, m_t$
- (3) Viết dạng tổng quát của nghiệm:

$$\begin{aligned} a_n = & (\alpha_{1,0} + \alpha_{1,1}n + \dots + \alpha_{1,m_1-1}n^{m_1-1})r_1^n \\ & + (\alpha_{2,0} + \alpha_{2,1}n + \dots + \alpha_{2,m_2-1}n^{m_2-1})r_2^n \\ & + \dots + (\alpha_{t,0} + \alpha_{t,1}n + \dots + \alpha_{t,m_t-1}n^{m_t-1})r_t^n \end{aligned}$$

- (4) Thay các điều kiện ban đầu $a_0 = C_0, a_1 = C_1, \dots, a_{k-1} = C_{k-1}$ vào dạng tổng quát để thiết lập hệ phương trình
- (5) Giải hệ phương trình để xác định các hệ số $\alpha_{i,j}$ với $1 \leq i \leq t$ và $0 \leq j \leq m_i - 1$
- (6) Thay các hệ số vào biểu thức tổng quát để có nghiệm cuối cùng

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

52

63

Giải hệ thức truy hồi

Đa thức đặc trưng



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Định lý 3: (Định lý 2 với $k = 2$)

Cho hệ thức truy hồi $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ ($n \geq 2$) với đa thức đặc trưng $r^2 - c_1 r - c_2$. Với điều kiện ban đầu $a_0 = C_0$, $a_1 = C_1$, dãy $\{a_n\}$ là nghiệm của hệ thức khi và chỉ khi:

(1) Khi đa thức đặc trưng có hai nghiệm phân biệt r_1, r_2 :

$$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$$

(2) Khi đa thức đặc trưng có một nghiệm bội 2 là r_1 :

$$a_n = (\alpha_1 + \alpha_2 n) r_1^n$$

với α_1, α_2 là các hằng số.

Giải hệ thức truy hồi

Đa thức đặc trưng



Ví dụ 20

Giải hệ thức truy hồi $f_n = f_{n-1} + f_{n-2}$ ($n \geq 2$) với điều kiện ban đầu $f_0 = 0$ và $f_1 = 1$

- Đa thức đặc trưng của hệ thức truy hồi là $r^2 - r - 1$
- Đa thức đặc trưng có hai nghiệm phân biệt $r_1 = \frac{1+\sqrt{5}}{2}$ và $r_2 = \frac{1-\sqrt{5}}{2}$
- Do đó, nếu dãy $\{f_n\}$ là nghiệm của hệ thức truy hồi thì $f_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ với các hằng số α_1, α_2 nào đó
- $\{f_n\}$ cần thỏa mãn điều kiện ban đầu $f_0 = 0$ và $f_1 = 1$. Thay vào dạng tổng quát của f_n , ta có hệ phương trình

$$f_0 = \alpha_1 + \alpha_2 = 0$$

$$f_1 = \alpha_1 \left(\frac{1+\sqrt{5}}{2} \right) + \alpha_2 \left(\frac{1-\sqrt{5}}{2} \right)$$

Từ đó $\alpha_1 = 1/\sqrt{5}$ và $\alpha_2 = -1/\sqrt{5}$

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

54

63

Giải hệ thức truy hồi

Đa thức đặc trưng



Ví dụ 21

Giải hệ thức truy hồi $a_n = 6a_{n-1} - 9a_{n-2}$ với các điều kiện ban đầu $a_0 = 1$ và $a_1 = 6$

- Đa thức đặc trưng của hệ thức truy hồi là $r^2 - 6r + 9$
- Đa thức đặc trưng có một nghiệm $r_1 = 3$ với bội 2
- Do đó, nếu dãy $\{a_n\}$ là nghiệm của hệ thức truy hồi thì $a_n = \alpha_{1,0}r_1^n + \alpha_{1,1}nr_1^n = \alpha_{1,0}3^n + \alpha_{1,1}n3^n$ với các hằng số $\alpha_{1,0}, \alpha_{1,1}$ nào đó
- Dãy $\{a_n\}$ cũng cần thỏa mãn điều kiện ban đầu $a_0 = 1$ và $a_1 = 6$. Thay vào dạng tổng quát của a_n , ta có hệ phương trình

$$a_0 = \alpha_{1,0} = 1$$

$$a_1 = \alpha_{1,0} \cdot 3 + \alpha_{1,1} \cdot 3 = 6$$

Do đó, ta có $\alpha_{1,0} = 1$ và $\alpha_{1,1} = 1$

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

55

63

Giải hệ thức truy hồi

Đa thức đặc trưng



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đệ quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

Định lý 4: (Định lý 2 với $k = 3$)

Cho hệ thức truy hồi $a_n = c_1 a_{n-1} + c_2 a_{n-2} + c_3 a_{n-3}$ ($n \geq 3$) với đa thức đặc trưng $r^3 - c_1 r^2 - c_2 r - c_3$. Với điều kiện ban đầu $a_0 = C_0$, $a_1 = C_1$, $a_2 = C_2$, dãy $\{a_n\}$ là nghiệm của hệ thức khi và chỉ khi:

(1) Khi đa thức đặc trưng có ba nghiệm phân biệt r_1, r_2, r_3 :

$$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \alpha_3 r_3^n$$

(2) Khi đa thức đặc trưng có hai nghiệm phân biệt r_1 (bội 2)

$$\text{và } r_2 \text{ (bội 1): } a_n = (\alpha_1 + \alpha_2 n) r_1^n + \alpha_3 r_2^n$$

(3) Khi đa thức đặc trưng có một nghiệm duy nhất r_1 (bội 3):

$$a_n = (\alpha_1 + \alpha_2 n + \alpha_3 n^2) r_1^n$$

trong đó $\alpha_1, \alpha_2, \alpha_3$ là các hằng số.

56

63

Giải hệ thức truy hồi

Đa thức đặc trưng



Ví dụ 22

Giải hệ thức truy hồi $a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$ với các điều kiện ban đầu $a_0 = 1$, $a_1 = -2$, và $a_2 = -1$

- Đa thức đặc trưng của hệ thức là $r^3 + 3r^2 + 3r + 1$
- Đa thức đặc trưng có một nghiệm $r_1 = -1$ với bội 3
- Do đó, nếu $\{a_n\}$ là nghiệm của hệ thức truy hồi thì $a_n = \alpha_{1,0}r_1^n + \alpha_{1,1}nr_1^n + \alpha_{1,2}n^2r_1^n$
- Dãy $\{a_n\}$ cũng cần thỏa mãn điều kiện ban đầu $a_0 = 1$, $a_1 = -2$, và $a_2 = -1$. Thay vào dạng tổng quát của a_n , ta có hệ phương trình

$$a_0 = \alpha_{1,0} = 1$$

$$a_1 = -\alpha_{1,0} - \alpha_{1,1} - \alpha_{1,2} = -2$$

$$a_2 = \alpha_{1,0} + 2\alpha_{1,1} + 4\alpha_{1,2} = -1$$

Do đó ta có $\alpha_{1,0} = 1$, $\alpha_{1,1} = 3$, và $\alpha_{1,2} = -2$

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

57

63

Giải hệ thức truy hồi

Đa thức đặc trưng



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy
nap sai

Đề quy

Định nghĩa và một số khái
niệm

Hàm định nghĩa bằng đệ
quy

Tập hợp định nghĩa bằng
đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy
hồi

Đoán nghiệm

Đa thức đặc trưng

Bài tập 23

Giải các hệ thức truy hồi với điều kiện ban đầu sau

(a) $a_n = 2a_{n-1}$ với $n \geq 1$, $a_0 = 3$

(b) $a_n = 5a_{n-1} - 6a_{n-2}$ với $n \geq 2$, $a_0 = 1$, $a_1 = 0$

(c) $a_n = 4a_{n-1} - 4a_{n-2}$ với $n \geq 2$, $a_0 = 6$, $a_1 = 8$

(d) $a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$ với $n \geq 3$, $a_0 = 5$, $a_1 = -9$,
 $a_2 = 15$

(e) $a_n = 6a_{n-1} - 8a_{n-2}$ ($n \geq 2$), $a_0 = 4$, $a_1 = 10$

(f) $a_n = 2a_{n-1} + a_{n-2} - 2a_{n-3}$ ($n \geq 3$), $a_0 = 3$, $a_1 = 6$, $a_2 = 0$

(g) (*) $\sqrt{a_n} = \sqrt{a_{n-1}} + 2\sqrt{a_{n-2}}$ ($n \geq 2$), $a_0 = a_1 = 1$

58

63

Giải hệ thức truy hồi

Đa thức đặc trưng



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

- **Hệ thức truy hồi tuyến tính không thuần nhất bậc k với hệ số hằng (Linear nonhomogeneous recurrence relation of degree k with constant coefficients)** là hệ thức có dạng

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n),$$

trong đó $c_1, c_2, \dots, c_k$ là các số thực, $c_k \neq 0$, $F(n)$ là một hàm chỉ phụ thuộc vào n và không phải luôn bằng 0, và $n \geq k$

- Hệ thức $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$ được gọi là **hệ thức truy hồi thuần nhất tương ứng (associated homogeneous recurrence relation)** của hệ thức trên

Định lý 5

Nếu $\{a_n^{(p)}\}$ là một nghiệm riêng nào đó của hệ thức $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n)$ thì mọi nghiệm của hệ thức đó có dạng $\{a_n^{(p)} + a_n^{(h)}\}$, trong đó $a_n^{(h)}$ là nghiệm của hệ thức thuần nhất tương ứng $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$

59

63

Giải hệ thức truy hồi

Đa thức đặc trưng



Với một số dạng $F(n)$, nghiệm riêng có dạng đặc biệt

Định lý 6

Giả sử $\{a_n\}$ thỏa mãn hệ thức

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n),$$

trong đó $c_1, \dots, c_k$ là các số thực, và

$$F(n) = (b_t n^t + b_{t-1} n^{t-1} + \cdots + b_1 n + b_0) s^n,$$

trong đó $b_0, \dots, b_t$ và s là các số thực. Khi s không phải là nghiệm của đa thức đặc trưng của hệ thức thuần nhất tương ứng, tồn tại một nghiệm riêng có dạng $(p_t n^t + p_{t-1} n^{t-1} + \cdots + p_1 n + p_0) s^n$. Khi s là một nghiệm với bội m của đa thức đặc trưng của hệ thức thuần nhất tương ứng, tồn tại một nghiệm riêng có dạng $n^m (p_t n^t + p_{t-1} n^{t-1} + \cdots + p_1 n + p_0) s^n$.

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

60

63

Giải hệ thức truy hồi

Đa thức đặc trưng



Các bước giải hệ thức truy hồi tuyến tính không thuần nhất bậc k với hệ số hằng

Cho hệ thức truy hồi: $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} + F(n)$ với $F(n) = (b_t n^t + \dots + b_1 n + b_0) s^n$

(1) Tìm nghiệm thuần nhất $a_n^{(h)}$:

- Xác định đa thức đặc trưng: $r^k - c_1 r^{k-1} - \dots - c_{k-1} r - c_k$
- Tìm nghiệm và bội tương ứng của đa thức đặc trưng
- Viết dạng tổng quát của nghiệm thuần nhất theo Định lý 2

(2) Tìm nghiệm riêng $a_n^{(p)}$:

- Kiểm tra xem s có phải là nghiệm đặc trưng của hệ thức thuần nhất hay không
- Nếu s không phải nghiệm đặc trưng: đặt $a_n^{(p)} = (p_t n^t + p_{t-1} n^{t-1} + \dots + p_1 n + p_0) s^n$
- Nếu s là nghiệm đặc trưng với bội m : đặt $a_n^{(p)} = n^m (p_t n^t + p_{t-1} n^{t-1} + \dots + p_1 n + p_0) s^n$
- Thay $a_n^{(p)}$ vào hệ thức ban đầu để thiết lập hệ phương trình tìm $p_0, p_1, \dots, p_t$

(3) Tìm nghiệm tổng quát: $a_n = a_n^{(h)} + a_n^{(p)}$

(4) Áp dụng điều kiện ban đầu để tìm các hằng số trong nghiệm thuần nhất

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Đề quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đề quy

Tập hợp định nghĩa bằng đề quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

61

63

Giải hệ thức truy hồi

Đa thức đặc trưng



Ví dụ 23

Giải hệ thức truy hồi $a_n = a_{n-1} + n$ ($n \geq 2$) với điều kiện ban đầu $a_1 = 1$

- Hệ thức thuần nhất tương ứng là $a_n = a_{n-1}$ ($n \geq 2$). Hệ thức này có nghiệm đặc trưng $r = 1$ và dãy $\{a_n^{(h)}\}$ với $a_n^{(h)} = c \cdot (1)^n$ là một dãy thỏa mãn hệ thức, trong đó c là hằng số nào đó
- Ta có $F(n) = n = (1 \cdot n + 0) \cdot 1^n$. Vì $s = 1$ là nghiệm đặc trưng của hệ thức thuần nhất tương ứng với bội 1, một nghiệm riêng của hệ thức không thuần nhất đã cho có dạng $a_n^{(p)} = n(p_1 n + p_0)1^n = p_1 n^2 + p_0 n$
- Thay dạng của nghiệm riêng vào hệ thức đã cho, ta có $n(2p_1 - 1) + (p_0 - p_1) = 0$, nghĩa là $2p_1 - 1 = 0$ và $p_0 - p_1 = 0$, do đó $p_0 = p_1 = 1/2$
- Cuối cùng, hệ thức ban đầu có nghiệm dạng $a_n = a_n^{(p)} + a_n^{(h)} = n(n+1)/2 + c$. Thay vào điều kiện ban đầu $a_1 = 1$ ta được $c = 0$

Quy nạp và Định quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy nạp sai

Định quy

Định nghĩa và một số khái niệm

Hàm định nghĩa bằng đệ quy

Tập hợp định nghĩa bằng đệ quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy hồi

Đoán nghiệm

Đa thức đặc trưng

62

63

Giải hệ thức truy hồi

Đa thức đặc trưng



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Giới thiệu

Quy nạp yếu

Quy nạp mạnh

Một số chứng minh quy
nap sai

Đề quy

Định nghĩa và một số khái
niệm

Hàm định nghĩa bằng đề
quy

Tập hợp định nghĩa bằng
đề quy

Quy nạp theo cấu trúc

Giải hệ thức truy hồi

Dãy cho bởi hệ thức truy
hồi

Đoán nghiệm

Đa thức đặc trưng

Bài tập 24

Giải các hệ thức truy hồi với điều kiện ban đầu sau

(1) $a_n = 3a_{n-1} + 2^n$ với $n \geq 1$, $a_0 = 1$

(2) $a_n = 2a_{n-1} + 2n^2$ với $n \geq 2$, $a_1 = 5$

(3) $a_n = -5a_{n-1} - 6a_{n-2} + 42 \cdot 4^n$ ($n \geq 3$), $a_1 = 56$, $a_2 = 278$

(4) $a_n = 5a_{n-1} - 6a_{n-2} + 2^n + 3n$ (**Gợi ý:** Tìm nghiệm riêng có dạng $qn2^n + p_1n + p_2$, trong đó q, p_1, p_2 là các hằng số)

Part I

Phụ lục

Nội dung



Quy nạp và Định quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy nạp mạnh

Quy nạp mạnh và quy nạp yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

Quy nạp toán học

Tính đúng đắn của Quy nạp mạnh

Quy nạp mạnh và quy nạp yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Quy nạp toán học

Tính đúng đắn của Quy nạp mạnh



Chứng minh (Quy nạp mạnh là đúng).

- Giả sử $P(1)$ đúng và với mọi $k \in \mathbb{Z}^+$, $(P(1) \wedge P(2) \wedge \dots \wedge P(k)) \rightarrow P(k+1)$ đúng. Ta chứng minh $\forall n \in \mathbb{Z}^+ P(n)$ bằng phản chứng
- Giả sử tồn tại $n \in \mathbb{Z}^+$ sao cho $P(n)$ sai. Do đó, tập $S = \{n \mid n \in \mathbb{Z}^+ \text{ và } P(n) \text{ sai}\} \subseteq \mathbb{Z}^+$ là tập khác rỗng.
- Theo Tiên đề 1, S có một phần tử nhỏ nhất m . Do $P(1)$ đúng và $m \in \mathbb{Z}^+$, ta có $m > 1$, suy ra $m - 1 \in \mathbb{Z}^+$
- Theo định nghĩa của m , với mọi số nguyên dương $j \leq m - 1$, ta có $P(j)$ đúng (nếu không thì $j < m$ và $P(j)$ sai; điều này mâu thuẫn với định nghĩa của m). Do đó $P(1) \wedge P(2) \wedge \dots \wedge P(m - 1)$ đúng
- Do $(P(1) \wedge P(2) \wedge \dots \wedge P(k)) \rightarrow P(k+1)$ đúng với mọi $k \in \mathbb{Z}^+$, ta có $(P(1) \wedge P(2) \wedge \dots \wedge P(m - 1)) \rightarrow P(m)$ đúng
- Kết hợp với $P(1) \wedge P(2) \wedge \dots \wedge P(m - 1)$ đúng, ta có $P(m)$ đúng. Điều này mâu thuẫn với định nghĩa của m . Do đó $P(n)$ đúng với mọi số nguyên dương n

Quy nạp và Đề quy
Hoàng Anh Đức

2

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi
Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

Quy nạp toán học

Quy nạp mạnh và quy nạp yếu là tương đương



Quy nạp yếu $\Rightarrow$ Quy nạp mạnh.

- (1) Giả sử với mọi $R(n)$, ta có
 $(R(1) \wedge \forall k \in \mathbb{Z}^+ (R(k) \rightarrow R(k+1))) \rightarrow \forall n \in \mathbb{Z}^+ R(n)$
đúng, nghĩa là, quy nạp yếu đúng
- (2) Giả sử với vị từ $P(n)$ ta có
 $P(1) \wedge \forall k \in \mathbb{Z}^+ \bigwedge_{j=1}^k P(j) \rightarrow P(k+1)$ đúng, nghĩa là,
bước cơ sở và bước quy nạp của quy nạp mạnh là đúng.
Ta chứng minh $\forall n \in \mathbb{Z}^+ P(n)$ đúng
- (3) Ta định nghĩa $Q(n) = \bigwedge_{j=1}^n P(j)$ với mọi $n \in \mathbb{Z}^+$
- (4) Từ (1), ta có
 $(Q(1) \wedge \forall k \in \mathbb{Z}^+ (Q(k) \rightarrow Q(k+1))) \rightarrow \forall n \in \mathbb{Z}^+ Q(n)$ đúng
- (5) Từ (3), ta có $Q(1) \equiv P(1)$ và $\forall n \in \mathbb{Z}^+ Q(n) \equiv \forall n \in \mathbb{Z}^+ P(n)$
- (6) Chú ý rằng với các mệnh đề p, q bất kỳ $p \rightarrow q \equiv p \rightarrow p \wedge q$
(Tại sao?).
- (7) Do đó $Q(k) \rightarrow Q(k+1) \equiv \bigwedge_{j=1}^k P(j) \rightarrow \bigwedge_{j=1}^{k+1} P(j) \equiv$
 $\bigwedge_{j=1}^k P(j) \rightarrow \bigwedge_{j=1}^k P(j) \wedge P(k+1) \equiv \bigwedge_{j=1}^k P(j) \rightarrow P(k+1)$
- (8) Thay (5) và (7) vào (4), ta có điều phải chứng minh

Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

3

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

Quy nạp toán học

Quy nạp mạnh và quy nạp yếu là tương đương



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

4

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

Quy nạp mạnh $\Rightarrow$ Quy nạp yếu.

- (1) Giả sử với mọi $R(n)$ ta có
 $(R(1) \wedge \forall k \in \mathbb{Z}^+ (\bigwedge_{j=1}^k R(j) \rightarrow R(k+1))) \rightarrow \forall n \in \mathbb{Z}^+ R(n)$
đúng, nghĩa là, quy nạp mạnh đúng
- (2) Giả sử với vị từ $P(n)$ ta có
 $P(1) \wedge \forall k \in \mathbb{Z}^+ (P(k) \rightarrow P(k+1))$ đúng, nghĩa là, bước cơ
sở và bước quy nạp của quy nạp yếu đúng. Ta chứng minh
 $\forall n \in \mathbb{Z}^+ P(n)$ đúng
- (3) Chú ý rằng với các mệnh đề p, q, r bất kỳ, nếu $p \rightarrow q$ đúng
thì $p \wedge r \rightarrow q$ cũng đúng (Tại sao?)
- (4) Áp dụng (1) với $P(n)$, ta có
 $(P(1) \wedge \forall k \in \mathbb{Z}^+ (\bigwedge_{j=1}^k P(j) \rightarrow P(k+1))) \rightarrow \forall n \in \mathbb{Z}^+ P(n)$
đúng.
- (5) Từ (2), ta có $P(1)$ đúng và $\forall k \in \mathbb{Z}^+ (P(k) \rightarrow P(k+1))$
đúng. Kết hợp với (3), ta có
 $\forall k \in \mathbb{Z}^+ (\bigwedge_{j=1}^k P(j) \rightarrow P(k+1))$ đúng
- (6) Từ (4) và (5) ta có điều phải chứng minh



Giải hệ thức truy hồi

Hàm sinh



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy nạp mạnh

Quy nạp mạnh và quy nạp yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

Hàm sinh

Hàm sinh (generating function) $G_a(x)$ của một dãy vô hạn $\{a_n\}$ ($n \geq 0$) được định nghĩa như sau

$$G_a(x) := \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k + \cdots$$

Nói cách khác, a_n là hệ số của x^n trong $G_a(x)$

$$\begin{aligned} G_a(x) &= \sum_{n=0}^{\infty} a_n x^n \\ a_n &= f(a_{n-1}, a_{n-2}, \dots) \\ a_0 &= C_0, a_1 = C_1, \dots \end{aligned}$$



Công thức tường minh
 $G_a(x) = h(x)$



Khái triển $h(x)$
 $h(x) = \sum_{n=0}^{\infty} \boxed{x^n}$

Giải hệ thức truy hồi

Hàm sinh



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

Định lý 7

Cho $f(x) = \sum_{n=0}^{\infty} a_n x^n$ và $g(x) = \sum_{n=0}^{\infty} b_n x^n$. Ta có

$$f(x) + g(x) = \sum_{n=0}^{\infty} (a_n + b_n) x^n$$

$$f(x)g(x) = \sum_{n=0}^{\infty} \left(\sum_{k=0}^n a_k b_{n-k} \right) x^n$$

Chú ý: Định lý trên chỉ đúng cho các chuỗi lũy thừa hội tụ trong một khoảng nào đó, và tất cả các chuỗi chúng ta sẽ xét trong bài giảng này đều thỏa mãn điều kiện đó. Tuy nhiên, ngay cả khi các chuỗi không hội tụ, *định lý trên có thể được sử dụng như là định nghĩa cho các phép cộng và nhân các hàm sinh*

Giải hệ thức truy hồi

Hàm sinh



Các bước giải hệ thức truy hồi bằng hàm sinh

(1) **Định nghĩa hàm sinh:** $G_a(x) = \sum_{n=0}^{\infty} a_n x^n$

(2) **Xây dựng phương trình cho $G_a(x)$:**

- Sử dụng hệ thức truy hồi, điều kiện ban đầu, và các phép biến đổi đại số để thu được phương trình đại số cho $G_a(x)$

(3) **Giải phương trình để tìm biểu thức tường minh $h(x)$ của $G_a(x)$**

(4) **Phân tích $h(x)$:**

- Khai triển $h(x)$ thành chuỗi lũy thừa sử dụng phân tích thành phân số riêng hoặc các công thức chuỗi cơ bản
- Áp dụng các đẳng thức đã biết để đưa về dạng chuỗi lũy thừa

(5) **Trích xuất hệ số:** So sánh với định nghĩa ban đầu

$$G_a(x) = \sum_{n=0}^{\infty} a_n x^n \text{ để xác định } a_n$$

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy nạp mạnh

Quy nạp mạnh và quy nạp yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

7

Giải hệ thức truy hồi

Hàm sinh



Ví dụ 24

Giải hệ thức truy hồi $a_n = 3a_{n-1}$ ($n \geq 1$) với điều kiện ban đầu $a_0 = 2$

$$\begin{aligned}G_a(x) &= \sum_{n=0}^{\infty} a_n x^n = a_0 + \sum_{n=1}^{\infty} (3a_{n-1}) x^n \\&= 2 + 3x \sum_{n=1}^{\infty} a_{n-1} x^{n-1} = 2 + 3x \sum_{m=0}^{\infty} a_m x^m \\&= 2 + 3x G_a(x) \\G_a(x) &= \frac{2}{1-3x}\end{aligned}$$

Nhắc lại rằng $\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$ với $-1 < x < 1$. Suy ra

$$\sum_{n=0}^{\infty} c^n x^n = \frac{1}{1-cx} \text{ với } -1 < cx < 1 \text{ trong đó } c \neq 0 \text{ là hằng số nào đó}$$

Sử dụng đẳng thức trên, ta có thể viết

$$G_a(x) = \frac{2}{1-3x} = 2 \sum_{n=0}^{\infty} 3^n x^n = \sum_{n=0}^{\infty} (2 \cdot 3^n) x^n$$

Suy ra $a_n = 2 \cdot 3^n$

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

8

23

Giải hệ thức truy hồi

Hàm sinh



Một phương pháp khác để tìm công thức tường minh cho $G_a(x)$ trong Ví dụ 24. Chú ý rằng $a_n = 3a_{n-1}$ ($n \geq 1$) và $a_0 = 2$

$$G_a(x) = \sum_{n=0}^{\infty} a_n x^n$$

$$-3xG_a(x) = -\sum_{n=0}^{\infty} [3a_n]x^{n+1}$$

$$G_a(x) - 3xG_a(x) = \sum_{n=0}^{\infty} a_n x^n - \sum_{n=0}^{\infty} [3a_n]x^{n+1}$$

$$= a_0 x^0 + \sum_{n=1}^{\infty} a_n x^n - \sum_{n=0}^{\infty} [3a_n]x^{n+1}$$

$$= 2 + \sum_{n=1}^{\infty} a_n x^n - \sum_{n=1}^{\infty} [3a_{n-1}]x^n$$

$$= 2 + \sum_{n=1}^{\infty} [a_n - 3a_{n-1}]x^n$$

$$= 2$$

Định nghĩa hàm sinh

Nhân hai vế với $-3x$

Cộng hai đẳng thức

Đổi chỉ số

$$a_n = 3a_{n-1}$$

Quy nạp và Định quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy nạp mạnh

Quy nạp mạnh và quy nạp yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

Giải hệ thức truy hồi

Hàm sinh



Quy nạp và Định quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

Một số đẳng thức hữu ích

$$\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n$$

$$\frac{1}{1-ax} = \sum_{n=0}^{\infty} a^n x^n$$

$$\frac{1}{1-x^k} = \sum_{n=0}^{\infty} x^{kn}$$

$$\frac{1}{(1-x)^2} = \sum_{n=0}^{\infty} (n+1)x^n$$

10

23

Giải hệ thức truy hồi

Hàm sinh



Ví dụ 25

Dãy Fibonacci $\{f_n\}$ cho bởi hệ thức $f_n = f_{n-1} + f_{n-2}$ ($n \geq 2$) và điều kiện ban đầu $f_0 = 0$ và $f_1 = 1$

$$G_f(x) = \sum_{n=0}^{\infty} f_n x^n = f_0 + f_1 x + \sum_{n=2}^{\infty} (f_{n-1} + f_{n-2}) x^n$$

$$= x + x \sum_{n=2}^{\infty} f_{n-1} x^{n-1} + x^2 \sum_{n=2}^{\infty} f_{n-2} x^{n-2}$$

$$= x + x \sum_{m=1}^{\infty} f_m x^m + x^2 \sum_{m=0}^{\infty} f_m x^m$$

$$= x + x \left[\sum_{m=0}^{\infty} f_m x^m - f_0 x^0 \right] + x^2 \sum_{m=0}^{\infty} f_m x^m$$

$$= x + x G_f(x) + x^2 G_f(x)$$

$$G_f(x) = \frac{x}{1 - x - x^2}$$

đổi biến

$$f_0 x^0 = 0$$

Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

11

23

Giải hệ thức truy hồi

Hàm sinh



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy nạp mạnh

Quy nạp mạnh và quy nạp yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

Một phương pháp khác để tìm công thức tường minh cho $G_f(x)$ trong ví dụ dãy Fibonacci. Chú ý rằng $f_n = f_{n-1} + f_{n-2}$ ($n \geq 2$) và $f_0 = 0, f_1 = 1$

$$G_f(x) = \sum_{n=0}^{\infty} f_n x^n$$

Định nghĩa hàm sinh

$$-xG_f(x) = -\sum_{n=0}^{\infty} f_n x^{n+1}$$

Nhân với $-x$

$$-x^2G_f(x) = -\sum_{n=0}^{\infty} f_n x^{n+2}$$

Nhân với $-x^2$

12

23

Giải hệ thức truy hồi

Hàm sinh



Quy nạp và Định quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy nạp mạnh

Quy nạp mạnh và quy nạp yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

Cộng ba đẳng thức

Đổi chỉ số

13

$$\begin{aligned} & G_f(x) - xG_f(x) - x^2G_f(x) \\ &= \sum_{n=0}^{\infty} f_n x^n - \sum_{n=0}^{\infty} f_n x^{n+1} - \sum_{n=0}^{\infty} f_n x^{n+2} \\ &= f_0 + f_1 x + \sum_{n=2}^{\infty} f_n x^n - \sum_{n=1}^{\infty} f_{n-1} x^n - \sum_{n=2}^{\infty} f_{n-2} x^n \\ &= 0 + x + \sum_{n=2}^{\infty} [f_n - f_{n-1} - f_{n-2}] x^n \\ &= x + 0 \\ &= x \end{aligned}$$

$$f_n = f_{n-1} + f_{n-2}$$

Do đó, $G_f(x)(1 - x - x^2) = x$, và suy ra $G_f(x) = \frac{x}{1 - x - x^2}$

Giải hệ thức truy hồi

Hàm sinh



Quy nạp và Đệ quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy nạp mạnh

Quy nạp mạnh và quy nạp yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

Nhắc lại rằng $\sum_{n=0}^{\infty} c^n x^n = \frac{1}{1-cx}$ với $-1 < cx < 1$ trong đó $c \neq 0$ là hằng số nào đó

Bài tập 25

- Viết $G_f(x) = \frac{x}{1-x-x^2} = \frac{a}{1-Ax} + \frac{b}{1-Bx}$ với các hằng số a, b, A, B nào đó
- Áp dụng công thức trên để đưa $G_f(x)$ về dạng $a \sum_{n=0}^{\infty} A^n x^n + b \sum_{n=0}^{\infty} B^n x^n$ với $-1 < Ax < 1$ và $-1 < Bx < 1$ ($\equiv$ Khai triển $G_f(x)$ thành chuỗi lũy thừa)
- Từ định nghĩa hàm sinh, suy ra công thức cho dãy $\{f_n\}$

14

23

Giải hệ thức truy hồi

Hàm sinh



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

Tài liệu tham khảo

Chú ý rằng từ $\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$ với $-1 < x < 1$, bằng cách lấy đạo hàm hai vế và đổi chỉ số lấy tổng, ta có

$$\sum_{n=1}^{\infty} nx^{n-1} = \sum_{m=0}^{\infty} (m+1)x^m = \frac{1}{(1-x)^2}$$

Bài tập 26

Sử dụng hàm sinh để giải các hệ thức truy hồi sau

- (a) $a_n = 7a_{n-1} \ (n \geq 1), a_0 = 5$
- (b) $a_n = 2a_{n-1} - a_{n-2} \ (n \geq 2), a_0 = 1, a_1 = 1$
- (c) $a_n = 5a_{n-1} - 6a_{n-2} \ (n \geq 2), a_0 = 6, a_1 = 30$
- (d) $(\star) a_n = 3a_{n-1} + n \ (n \geq 1), a_0 = 1$

15

23

Một số lỗi thường gặp



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

16

Một số lỗi thường gặp

Tài liệu tham khảo

Chú ý

Tham khảo từ tài liệu “Common Mistakes in Discrete Mathematics” (https://highereducation.com/sites/default/files/125967651x/1106131/Common_Mistakes_in_Discrete_Math.pdf)

(a) Quên thực hiện bước cơ sở trong chứng minh bằng quy nạp toán học

- Bước quy nạp có thể thực hiện tốt, ví dụ, khi chúng ta cố gắng chứng minh rằng $n = n + 1$ với mọi số nguyên dương n , nhưng mệnh đề này rõ ràng là không đúng. Vấn đề là ở bước cơ sở (khi $n = 1$) không thỏa mãn, vì $1 \neq 1 + 1$

(b) Không thực hiện nhiều hơn một trường hợp trong bước cơ sở của chứng minh bằng quy nạp

Một số lỗi thường gặp (tiếp)



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy nạp mạnh

Quy nạp mạnh và quy nạp yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

17

Một số lỗi thường gặp

Tài liệu tham khảo

- Trong một số tình huống, chẳng hạn như khi bước quy nạp cần hai hoặc nhiều điều kiện trước đó, cần phải kiểm tra nhiều trường hợp cơ sở
- Ví dụ, khi chứng minh các phát biểu về dãy Fibonacci, thường cần phải kiểm tra hai trường hợp cơ sở đầu tiên (giả sử $n = 1$ và $n = 2$), vì bước quy nạp dựa vào phương trình $f_n = f_{n-1} + f_{n-2}$

(c) Nhầm lẫn giữa tổng và vị từ $P(n)$ trong chứng minh quy nạp

- Ví dụ, khi cố gắng chứng minh $1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$ bằng quy nạp, $P(n)$ là toàn bộ phương trình này, không phải chỉ là vế trái của nó

(d) Viết $P(k+1)$ không chính xác trong chứng minh bằng quy nạp toán học

- Khi $P(n)$ đã được xác định đúng, việc viết $P(k+1)$ thường có thể được thực hiện một cách máy móc bằng cách thay $k+1$ vào n

Một số lỗi thường gặp (tiếp)



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

18 Một số lỗi thường gặp

Tài liệu tham khảo

- Ví dụ, nếu $P(n)$ là mệnh đề $2 + 4 + 6 + \cdots + 2n = n(n + 1)$, thì $P(k + 1)$ là $2 + 4 + 6 + \cdots + 2k + 2(k + 1) = (k + 1)(k + 2)$

(e) Rơi vào “bẫy quy nạp (induction trap)” trong chứng minh bằng quy nạp toán học

- Đây là thuật ngữ được Douglas West (Đại học Illinois) đặt ra để chỉ lỗi khi bắt đầu từ tính đúng đắn của $P(k)$ và chứng minh tính đúng đắn của $P(k + 1)$ chỉ trong một số trường hợp đặc biệt, thay vì chứng minh tổng quát

(f) Mắc lỗi khi thực hiện các biến đổi đại số cơ bản khi chứng minh bằng quy nạp toán học

- Có thể xảy ra trong khi thực hiện các biến đổi đại số cơ bản, đặc biệt là trong việc đơn giản hóa biểu thức
- Ví dụ, bạn có thể cần sử dụng đẳng thức $2^n + 2^n = 2^{n+1}$, hoặc cần đơn giản hóa $(n + 1)^3 + 5(n + 1)^2$, điều này tốt nhất nên được thực hiện bằng cách phân tích thừa số, không phải bằng cách khai triển từng số hạng

Một số lỗi thường gặp (tiếp)



Quy nạp và Định quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

19

Một số lỗi thường gặp

Tài liệu tham khảo

- Kiểm tra kỹ các thao tác biến đổi đại số của bạn khi bạn gặp khó khăn với bước quy nạp của một chứng minh như vậy

(g) Thiểu tổ chức khi viết định nghĩa đệ quy

- Một lời khuyên là nên suy nghĩ về cách xây dựng từng bước một các đối tượng đang xét
- Các quy tắc ở bước đệ quy của định nghĩa cần được xây dựng để cho phép từng bước như vậy, và các trường hợp cơ sở là cần thiết để bắt đầu quá trình xây dựng đối tượng
- Những lỗi phổ biến bao gồm:
 - Không đưa đủ các trường hợp cơ sở (ví dụ, định nghĩa đệ quy của dãy Fibonacci yêu cầu hai trường hợp cơ sở)
 - Có các trường hợp xung đột (ví dụ, có một trường hợp cho n chia hết cho 2 và một trường hợp khác cho n chia hết cho 3, dẫn đến không có định nghĩa duy nhất cho các trường hợp n chia hết cho 6)
 - Có giá trị hàm tại n phụ thuộc vào giá trị hàm tại đầu vào lớn hơn n (ví dụ, $f(n) = f(3n + 1) - 2$)

Một số lỗi thường gặp (tiếp)



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy nạp mạnh

Quy nạp mạnh và quy nạp yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

20 Một số lỗi thường gặp

Tài liệu tham khảo

(h) Dễ dàng từ bỏ khi cố gắng viết một quan hệ truy hồi để mô hình hóa một bài toán

- Hãy tự hỏi bản thân làm thế nào để có thể nhận được một trường hợp của bài toán kích thước n từ các trường hợp kích thước $n - 1$ (hoặc đôi khi là các trường hợp nhỏ hơn)
- Đảm bảo xem xét tất cả các khả năng, và nhớ bao gồm đủ các điều kiện ban đầu
- Ví dụ, nếu a_n là số cách để leo n bậc thang nếu chúng ta được phép leo mỗi lần một bậc hoặc ba bậc, thì rõ ràng $a_1 = 1$, $a_2 = 1$, và $a_3 = 2$, và sau đó $a_n = a_{n-1} + a_{n-3}$ với $n \geq 4$, vì bước đầu tiên có thể là một bước đơn hoặc ba bước

(i) Áp dụng sai thuật toán giải quan hệ truy hồi tuyến tính thuần nhất với hệ số hằng khi phương trình đặc trưng có nghiệm lặp

- Trong trường hợp này, cần phải nhân với lũy thừa của n

Một số lỗi thường gặp (tiếp)



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy nạp mạnh

Quy nạp mạnh và quy nạp yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

21 Một số lỗi thường gặp

Tài liệu tham khảo

- Ví dụ, nếu phương trình đặc trưng là

$$r^2 - 6r + 9 = (r - 3)^2 = 0, \text{ thì nghiệm tổng quát là}$$

$$a_n = c_1 \cdot 3^n + c_2 \cdot n \cdot 3^n$$

(j) Tìm nghiệm riêng không chính xác của quan hệ truy hồi tuyến tính không thuần nhất với hệ số hằng

- Luôn nên kiểm tra lại các nghiệm bạn tìm được. Ví dụ, nếu bạn tính được rằng $a_n = 2^n$ là nghiệm riêng của

$$a_n = 2a_{n-1} + 2^n, \text{ thì khi thế vào, bạn sẽ thấy mình đã mắc lỗi, vì không đúng rằng } 2^n = 2 \cdot 2^{n-1} + 2^n$$

(k) Quên quan tâm đến vài số hạng đầu tiên của chuỗi lũy thừa

- Khi giải quan hệ truy hồi bằng cách sử dụng hàm sinh, quan hệ truy hồi thường chỉ áp dụng được cho $k \geq 1$ hoặc $k \geq 2$; do đó một hoặc hai số hạng đầu tiên phải được xử lý riêng biệt

(l) Quên đổi biến trong chuỗi lũy thừa khi cần thiết

Một số lỗi thường gặp (tiếp)



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

22 Một số lỗi thường gặp

Tài liệu tham khảo

- Ví dụ, nếu chuỗi lũy thừa có x^{k-1} và bạn cần chuyển nó thành x^k , bạn có thể thay k bằng $k+1$ trong toàn bộ tổng (bao gồm cả các giới hạn) và đơn giản hóa về mặt đại số:

$$\sum_{k=1}^{\infty} kx^{k-1} = \sum_{k+1=1}^{\infty} (k+1)x^{(k+1)-1} = \sum_{k=0}^{\infty} (k+1)x^k$$



Quy nạp và Đề quy

Hoàng Anh Đức

Quy nạp toán học

Tính đúng đắn của Quy
nap mạnh

Quy nạp mạnh và quy nạp
yếu là tương đương

Giải hệ thức truy hồi

Hàm sinh

Một số lỗi thường gặp

23

Tài liệu tham khảo



Gunderson, David S. and Kenneth H. Rosen (2010).
*Handbook of Mathematical Induction: Theory and
Applications*. Chapman and Hall/CRC. DOI:
10.1201/b16005.

1. Logic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Độ quy
4. Thuật toán I: Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp
5. Thuật toán II: Thuật toán tham lam, thuật toán đệ quy
6. Lý thuyết số cơ bản
7. Các phương pháp đếm
8. Lý thuyết đồ thị I: Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông
9. Lý thuyết đồ thị II: Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị
10. Lý thuyết đồ thị III: Cây
11. Đại số Boole

VNU-HUS MAT3500: Toán rời rạc

Thuật toán I

Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn



Nội dung



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Định nghĩa và một số khái niệm



Thuật toán I

Hoàng Anh Đức

- Một **thuật toán (algorithm)** là một tập hữu hạn các hướng dẫn cụ thể để thực hiện một nhiệm vụ nào đó
 - cộng hai số tự nhiên biểu diễn dưới dạng số thập phân
 - đăng ký môn học trực tuyến
 - đi từ nhà đến trường
 - Một **chương trình máy tính (computer program)** là
 - một mô tả của thuật toán nào đó
 - sử dụng một ngôn ngữ đủ chuẩn xác để máy tính có thể hiểu
 - cùng với các phép toán mà máy tính đã biết cách thực hiện
- Ta nói rằng thuật toán được **cài đặt (implement)** cụ thể bằng chương trình máy tính
- Khi mở một phần mềm trong máy tính, ta nói rằng **chương trình hoặc thuật toán của nó được chạy hoặc được thực hiện bởi máy tính**
 - Khi có mô tả của một thuật toán, bạn cũng **có thể thực hiện từng bước của thuật toán với giấy và bút**

2

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Định nghĩa và một số khái niệm



Thuật toán I
Hoàng Anh Đức

Một số tính chất của một thuật toán

Đầu vào (Input) Một thuật toán có các giá trị đầu vào từ một tập đã được xác định trước

Đầu ra (Output) Từ mỗi một tập các giá trị đầu vào, một thuật toán sinh ra các giá trị đầu ra. Các giá trị này chính là lời giải cho bài toán

Tính xác định (Definiteness) Các bước của một thuật toán cần phải được xác định một cách chính xác

Tính đúng đắn (Correctness) Với mỗi tập giá trị đầu vào, một thuật toán cần cho ra kết quả đầu ra đúng

Tính hữu hạn (Finiteness) Với mỗi tập giá trị đầu vào, một thuật toán cần cho ra các giá trị đầu ra mong muốn sau một số hữu hạn (có thể là rất lớn) các bước

Tính hiệu quả (Effectiveness) Mỗi bước của thuật toán cần được thực hiện một cách chính xác và trong thời gian hữu hạn

Tính tổng quát (Generality) Thuật toán phải áp dụng được cho mọi bài toán mong muốn, chứ không phải chỉ với một tập các giá trị đầu vào đặc biệt

3 Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Định nghĩa và một số khái niệm



Thuật toán I
Hoàng Anh Đức

Một *mô tả đầy đủ của một thuật toán* bao gồm ba phần

(1) *Thuật toán (algorithm)*

- Mô tả một cách rõ ràng và chính xác nhất có thể
 - Một *thuật toán* có thể được *mô tả bằng một ngôn ngữ máy tính* (C, Python, Java, v.v...). Tuy nhiên, những mô tả này cần tuân theo các chỉ dẫn cụ thể trong ngôn ngữ máy tính tương ứng. Điều này dẫn đến việc các mô tả theo phương pháp này thường phức tạp và khó hiểu
 - Thay vì dùng một ngôn ngữ máy tính cụ thể để mô tả thuật toán, ta sử dụng *ngôn ngữ thông thường (natural language)*, *giả mã (pseudocode)*, hoặc *sơ đồ khối (flowchart)*
- Thường kèm theo mô tả ngắn gọn về ý tưởng của thuật toán

(2) Một chứng minh về *tính đúng đắn (correctness)* của thuật toán

- Với mọi tập đầu vào hợp lệ, thuật toán cần cho kết quả đầu ra đúng

(3) Một phân tích về *hiệu năng (performance)* của thuật toán

- Thời gian thực thi, không gian bộ nhớ, v.v...

4

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường



Thuật toán I

Hoàng Anh Đức

Sử dụng *ngôn ngữ thông thường (natural language)* để mô tả từng bước thực hiện thuật toán

Bước 1 Thực hiện việc X **Bước 3** Lặp lại Z

Bước 2 Tính Y **Bước 4** ...

Ví dụ 1 (Mô tả bằng ngôn ngữ thông thường)

■ Bài toán:

- **Input:** Dãy số nguyên $a_1, a_2, \dots, a_n$
- **Output:** Giá trị của phần tử lớn nhất trong dãy

■ Tìm giá trị của phần tử lớn nhất:

Bước 1 Gán biến v (lưu giá trị lớn nhất hiện tại) bằng a_1

Bước 2 Lần lượt xét các phần tử $a_2, a_3, \dots, a_n$:

- Nếu phần tử đang xét lớn hơn v , cập nhật v bằng giá trị của phần tử đó
- Ngược lại, giữ nguyên giá trị của v

Bước 3 Sau khi xét hết tất cả các phần tử, trả về giá trị của v (chính là phần tử lớn nhất trong dãy)

Định nghĩa và một số khái niệm

Mô tả thuật toán

5

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

6

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Ví dụ 2 (Thực hiện thuật toán)

■ **Input:** Dãy $a_1 = 7, a_2 = 12, a_3 = 5, a_4 = 16, a_5 = 9$

■ **Output:** Giá trị của phần tử lớn nhất trong dãy

a_1	a_2	a_3	a_4	a_5
7	12	5	16	9
	$i = 2$	$i = 3$	$i = 4$	$i = 5$
$v = 7$	$v = 12$	$v = 12$	$v = 16$	$v = 16$

Mô tả thuật toán

Mô tả bằng giả mã



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

- **Giả mã (pseudocode)** là một dạng **hỗn hợp giữa ngôn ngữ thông thường và ngôn ngữ lập trình**
- Một **biến (variable)** được sử dụng để biểu diễn vị trí trong bộ nhớ máy tính để lưu trữ một giá trị. Khi ta nói đến một biến X nào đó, trên thực tế, chúng ta muốn sử dụng giá trị lưu tại một vị trí nào đó trong bộ nhớ ứng với X

Phép gán (assignment)

$\text{variable} := \text{expression}$

- Tính toán **expression**
- Lưu kết quả vào (vị trí trong bộ nhớ ứng với) biến **variable**
- Chú ý rằng $=$ và $:=$ khác nhau. Do đó, trong nhiều tài liệu, ký hiệu $\leftarrow$ được sử dụng thay thế cho $:=$.

Mô tả thuật toán

Mô tả bằng giả mã



Thuật toán I

Hoàng Anh Đức

Cấu trúc điều kiện (conditional statement)

```
if condition then
    S1
else
    S2
```

- Tính **condition**; trả lại True hoặc False
- Nếu True, thực hiện S_1 ; nếu False, thực hiện S_2
- Sau đó tiếp tục với các lệnh tiếp theo sau cấu trúc điều kiện

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Mô tả thuật toán

Mô tả bằng giả mã



Thuật toán I

Hoàng Anh Đức

Vòng lặp **for** (**for loop**)

```
for variable := initial_value to final_value do  
    S
```

- Khởi tạo **variable** với giá trị **initial_value**
- Nếu **variable** $\leq$ **final_value**, thực hiện đoạn mã **S**
- Sau mỗi lần lặp, tăng **variable** thêm 1
- Kiểm tra lại điều kiện; nếu vẫn thỏa mãn, tiếp tục lặp
- Khi **variable** $>$ **final_value**, thoát vòng lặp, thực hiện các lệnh tiếp theo ngay sau vòng **for**

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Mô tả thuật toán

Mô tả bằng giả mã



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Vòng lặp **while** (*while loop*)

while condition do
S

- Kiểm tra giá trị của **condition**
- Nếu **condition** là **True**, thực hiện đoạn mã *S*
- Sau khi thực hiện *S*, kiểm tra lại **condition**
- Nếu vẫn **True**, thực hiện *S* lần nữa
- Khi **condition** trở thành **False**, thoát khỏi vòng lặp và thực hiện các lệnh tiếp theo ngay sau vòng **while**

10

66

Mô tả thuật toán

Mô tả bằng giả mã



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Vòng lặp **do-while** (**do-while** loop)

```
do
    S
while condition
```

- Thực hiện đoạn mã S trước
- Sau đó, kiểm tra giá trị của **condition**
- Nếu **condition** là True, thực hiện S lần nữa
- Tiếp tục lặp lại quá trình cho đến khi **condition** trở thành False
- Khác với vòng lặp **while**, vòng lặp **do-while** luôn thực hiện S ít nhất một lần
- Một số giáo trình đề cập cấu trúc **repeat...until** tương tự, với **until** sử dụng điều kiện ngược lại để dừng lặp

11

66

Mô tả thuật toán

Mô tả bằng giả mã



Thuật toán I

Hoàng Anh Đức

Lệnh trả về (*return statement*)

`return expression`

- Trả về giá trị của `expression` làm kết quả đầu ra của thuật toán
- Khi gặp lệnh **return**, thuật toán sẽ kết thúc ngay lập tức
- Nếu **return** nằm trong một hàm, nó sẽ trả về giá trị cho nơi gọi hàm và kết thúc việc thực thi hàm

Nhận xét và chú thích (*comments*)

- Nhận xét là văn bản giải thích mà chương trình sẽ bỏ qua khi thực thi
- Được sử dụng để làm rõ code hoặc giải thích logic
- Các dạng phổ biến:
 - `// Nhận xét một dòng`
 - `/* Nhận xét nhiều dòng */`
 - `⟪ Nhận xét ⟫`

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

12

66

Mô tả thuật toán

Mô tả bằng giả mã



Thuật toán I

Hoàng Anh Đức

Ví dụ 3 (Mô tả thuật toán bằng giả mã)

Thuật toán 1: Tìm giá trị của phần tử lớn nhất

Input: $a_1, a_2, \dots, a_n$: dãy số nguyên

Output: Giá trị của phần tử lớn nhất trong dãy

```
1  $v := a_1$  // phần tử lớn nhất đến hiện tại
2 for  $i := 2$  to  $n$  do // lần lượt xét  $a_2, \dots, a_n$ 
3     if  $a_i > v$  then //  $a_i >$  phần tử lớn nhất hiện tại?
4          $v := a_i$  // bây giờ  $v$  lớn nhất trong
            $a_1, \dots, a_i$ 
5 return  $v$ 
```

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt buộc vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Mô tả thuật toán

Mô tả bằng sơ đồ khối



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

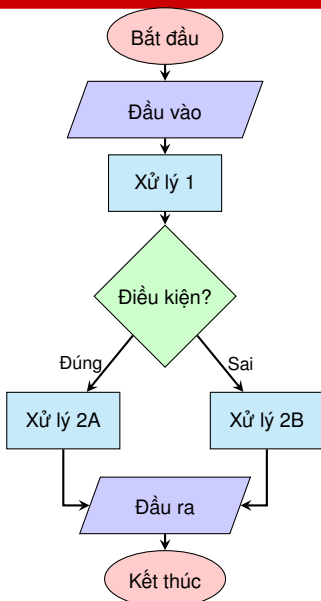
Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

- **Sơ đồ khối (flowchart)** là một loại biểu đồ biểu diễn một luồng công việc hoặc quy trình
- Hiển thị các bước dưới dạng các hộp hình dạng khác nhau
- Thứ tự thực hiện được kết nối bằng các mũi tên
- Các thành phần chính:
 - **Bắt đầu/Kết thúc** (hình elip)
 - **Xử lý** (hình chữ nhật)
 - **Điều kiện** (hình thoi)
 - **Đầu vào/Đầu ra** (hình thang)



14

66

Mô tả thuật toán

Mô tả bằng sơ đồ khối



Ví dụ 4 (Mô tả thuật toán bằng sơ đồ khối)

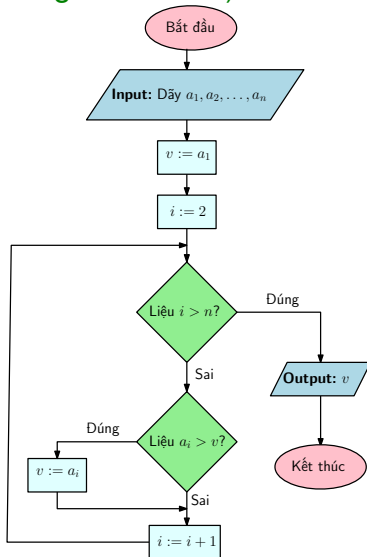
Bài toán:

- **Input:** Dãy số nguyên

$a_1, a_2, \dots, a_n$

- **Output:** Giá trị của phần tử lớn nhất trong dãy

Ý tưởng: Duyệt qua từng phần tử của dãy, lưu giữ giá trị lớn nhất đã tìm thấy cho đến hiện tại.



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

15

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt đầu vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

66

Chứng minh thuật toán

Bắt biên vòng lặp



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng đồ khởi

Chứng minh thuật toán

16 Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

66

- Có rất nhiều phương pháp khác nhau để *chứng minh tính đúng đắn của một thuật toán*
- Một trong số đó là sử dụng *bắt biên vòng lặp (loop invariant)*—một phương pháp được xây dựng dựa trên phương pháp quy nạp toán học
 - *Vòng lặp (loop): for, while, v.v...*
 - Một *bắt biên vòng lặp* là *một phát biểu luôn đúng trước và sau mỗi lần lặp (iteration) của một vòng lặp (loop)*

What can I learn right now in just 10 minutes that could improve my algorithmic thinking?

All related (92) ▾

Sort Recommended ▾



Thomas Cormen

The C in CLRS. - Featured on ForbesUpvoted by Vaibhav Krishan, Ph. D. student Computer Science & Computational Complexity Theory, Indian Institute of Technology, Bombay (20... and Wei-Cheng Lin, M.S. Computer Science, National Taiwan University (2016)Author has **842** answers and **40.7M** answer views · 10y

Originally Answered: What can I learn right now in just 10 minutes that could improve my algorithmic thinking ?

It's pretty hard to answer that question without knowing what you already know. If I had to give just one thing, that thing would be loop invariants. Understand that when you write a loop, you either implicitly or explicitly use a loop invariant.

Chứng minh thuật toán

Bất biến vòng lặp



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

17 Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

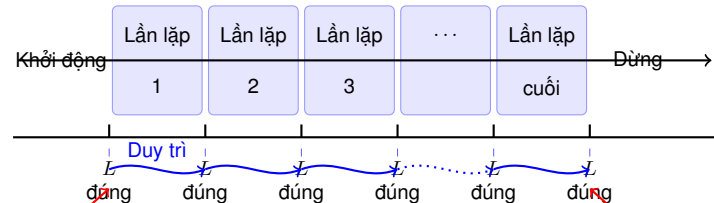
Sắp xếp chèn

Một bất biến vòng lặp L là một phát biểu thỏa mãn các tính chất sau:

Khởi động (Initialization) L đúng trước lần lặp đầu tiên của vòng lặp

Duy trì (Maintenance) Nếu L đúng trước một lần lặp của vòng lặp thì nó cũng đúng trước lần lặp tiếp theo

Dừng (Termination) Khi vòng lặp dừng, bất biến vòng lặp cho ta một tính chất hữu ích để chứng minh thuật toán đúng



Khởi động

Kết quả đúng

Chứng minh thuật toán

Bất biến vòng lặp



Thuật toán 1

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

18

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

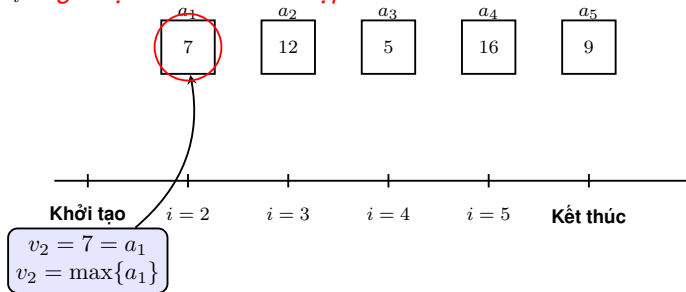
Sắp xếp chèn

Ví dụ 5

Một bất biến vòng lặp trong Thuật toán 1 (vòng **for** ở Dòng 2–4) tìm giá trị lớn nhất trong dãy số nguyên $a_1, \dots, a_n$

$L = \text{“Ở trước lần lặp với biến } i, v = \max\{a_1, a_2, \dots, a_{i-1}\} \text{”}$

Gọi v_i là *giá trị của v trước lần lặp với biến i*



Khởi động: $v_2 = a_1 = \max\{a_1\}$

Chứng minh thuật toán

Bất biến vòng lặp



Thuật toán 1

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

18

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

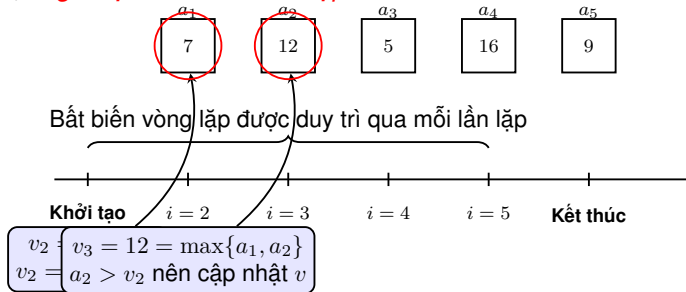
Sắp xếp chèn

Ví dụ 5

Một bất biến vòng lặp trong Thuật toán 1 (vòng **for** ở Dòng 2–4) tìm giá trị lớn nhất trong dãy số nguyên $a_1, \dots, a_n$

$L = \text{“Ở trước lần lặp với biến } i, v = \max\{a_1, a_2, \dots, a_{i-1}\} \text{”}$

Gọi v_i là *giá trị của v trước lần lặp với biến i*



Khởi động: $v_2 = a_1 = \max\{a_1\}$

Trước $i = k$: $v_k = \max\{a_1, \dots, a_{k-1}\}$

Duy trì: Trước $i = k + 1$: $v_{k+1} = \max\{a_1, \dots, a_k\}$

66

Chứng minh thuật toán

Bất biến vòng lặp

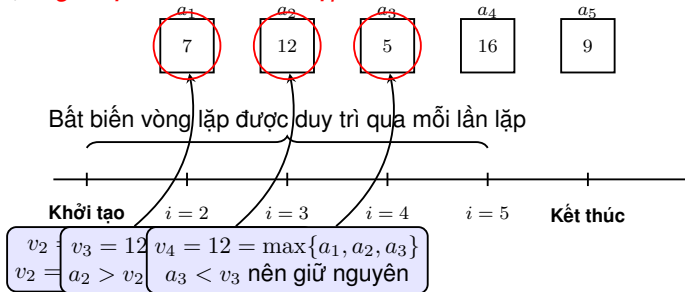


Ví dụ 5

Một bất biến vòng lặp trong Thuật toán 1 (vòng **for** ở Dòng 2–4) tìm giá trị lớn nhất trong dãy số nguyên $a_1, \dots, a_n$

$L =$ “*Ở trước lần lặp với biến i , $v = \max\{a_1, a_2, \dots, a_{i-1}\}$* ”

Gọi v_i là *giá trị của v trước lần lặp với biến i*



Khởi động: $v_2 = a_1 = \max\{a_1\}$

Trước $i = k$: $v_k = \max\{a_1, \dots, a_{k-1}\}$

Duy trì: Trước $i = k + 1$: $v_{k+1} = \max\{a_1, \dots, a_k\}$

Thuật toán 1

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

18

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Chứng minh thuật toán

Bất biến vòng lặp



Thuật toán 1

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

18

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

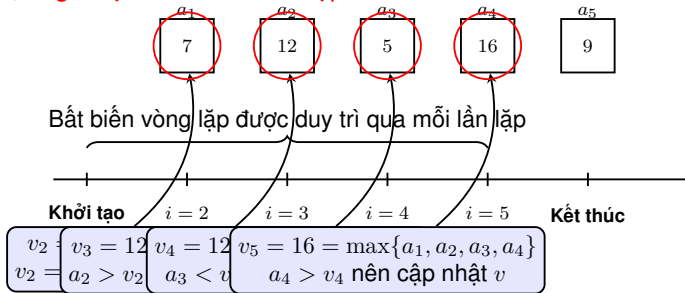
Sắp xếp chèn

Ví dụ 5

Một bất biến vòng lặp trong Thuật toán 1 (vòng **for** ở Dòng 2–4) tìm giá trị lớn nhất trong dãy số nguyên $a_1, \dots, a_n$

$L = \text{“Ở trước lần lặp với biến } i, v = \max\{a_1, a_2, \dots, a_{i-1}\} \text{”}$

Gọi v_i là *giá trị của v trước lần lặp với biến i*



Khởi động: $v_2 = a_1 = \max\{a_1\}$

Trước $i = k$: $v_k = \max\{a_1, \dots, a_{k-1}\}$

Duy trì: Trước $i = k + 1$: $v_{k+1} = \max\{a_1, \dots, a_k\}$

Chứng minh thuật toán

Bất biến vòng lặp



Thuật toán 1

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

18

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

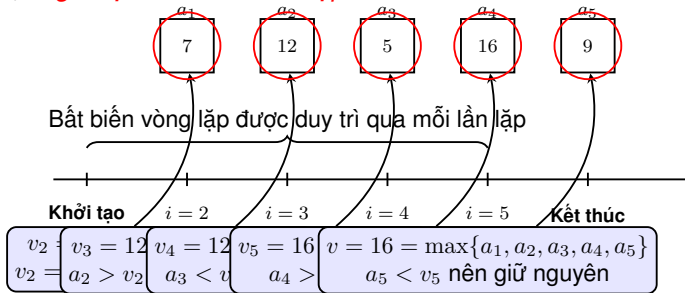
Sắp xếp chèn

Ví dụ 5

Một bất biến vòng lặp trong Thuật toán 1 (vòng **for** ở Dòng 2–4) tìm giá trị lớn nhất trong dãy số nguyên $a_1, \dots, a_n$

$L = \text{“Ở trước lần lặp với biến } i, v = \max\{a_1, a_2, \dots, a_{i-1}\}”$

Gọi v_i là *giá trị của v trước lần lặp với biến i*



Khởi động: $v_2 = a_1 = \max\{a_1\}$

Trước $i = k$: $v_k = \max\{a_1, \dots, a_{k-1}\}$

Duy trì: Trước $i = k + 1$: $v_{k+1} = \max\{a_1, \dots, a_k\}$

Chứng minh thuật toán

Bất biến vòng lặp



Thuật toán 1

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

18 Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

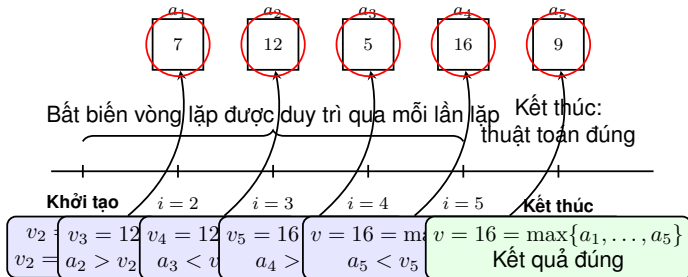
Sắp xếp chèn

Ví dụ 5

Một bất biến vòng lặp trong Thuật toán 1 (vòng **for** ở Dòng 2–4) tìm giá trị lớn nhất trong dãy số nguyên $a_1, \dots, a_n$

$L = \text{“Ở trước lần lặp với biến } i, v = \max\{a_1, a_2, \dots, a_{i-1}\}”$

Gọi v_i là *giá trị của v trước lần lặp với biến i*



Khởi động: $v_2 = a_1 = \max\{a_1\}$

Trước $i = k$: $v_k = \max\{a_1, \dots, a_{k-1}\}$

Duy trì: Trước $i = k + 1$: $v_{k+1} = \max\{a_1, \dots, a_k\}$

Dừng: $v = \max\{a_1, \dots, a_n\}$

Chứng minh thuật toán

Bất biến vòng lặp



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

19 Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Chứng minh tính chất của bất biến vòng lặp.

$L = \text{"Ở trước lần lặp với biến } i, v = \max\{a_1, a_2, \dots, a_{i-1}\}"$

Gọi v_i là giá trị của v trước lần lặp với biến i

- **Khởi động** ($i = 2$): Ta cần chỉ ra rằng trước vòng **for**,
 $v_2 = \max\{a_1\} = a_1$, và điều này hiển nhiên đúng do Thuật toán 1 gán v bằng a_1 ở Dòng 1
- **Duy trì**: Giả sử L đúng ở trước lần lặp với $i = k$ nào đó, nghĩa là
 $v_k = \max\{a_1, \dots, a_{k-1}\}$. Ta chứng minh L đúng ở trước lần lặp với
 $i = k + 1$, nghĩa là $v_{k+1} = \max\{a_1, \dots, a_{k-1}, a_k\}$. Ta xét các trường hợp dựa trên điều kiện ở Dòng 3
 - Nếu $a_k > v = v_k$ sai, giá trị của v không thay đổi, và do đó $v_{k+1} = v_k$. Ta có $\max\{a_1, \dots, a_{k-1}, a_k\} = \max\{v_k, a_k\} = v_k$. Suy ra
 $v_{k+1} = \max\{a_1, \dots, a_{k-1}, a_k\}$
 - Nếu $a_k > v = v_k$ đúng, giá trị của v được gán bằng a_k , và do đó
 $v_{k+1} = a_k$. Ta cũng có $\max\{a_1, \dots, a_{k-1}, a_k\} = \max\{v_k, a_k\} = a_k$. Suy ra
 $v_{k+1} = \max\{a_1, \dots, a_{k-1}, a_k\}$
- **Dừng**: Sau khi kết thúc lần lặp $i = n$ (hoặc, trước khi bắt đầu lần lặp $i = n + 1$ mà sẽ không bao giờ được thực hiện), $v = \max\{a_1, \dots, a_n\}$ và do đó là giá trị lớn nhất của các phần tử trong dãy đầu vào



Chứng minh thuật toán

Bất biến vòng lặp



Thuật toán I

Hoàng Anh Đức

Bài tập 1

Một thuật toán tính x^n với $x \in \mathbb{R}^+$ và $n \in \mathbb{N}$ được mô tả như sau

Thuật toán 2: Tính x^n .

Input: x : số thực dương, n : số tự nhiên

Output: Giá trị của x^n

```
1 answer := 1
2 m := n
3 while m > 0 do
4   |   answer := answer · x
5   |   m := m - 1
6 return answer
```

Hãy chứng minh phát biểu L sau là một bất biến vòng lặp cho vòng **while**

$L =$ “Ở trước lần lặp với biến m , $\textit{answer} = x^{n-m}$ ”

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

20 Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

- **Phân tích thuật toán (algorithm analysis)** nghiên cứu cách đánh giá **hiệu quả** của thuật toán bằng cách ước lượng lượng **thời gian (time)** và **bộ nhớ (space)** cần thiết
- **Độ phức tạp (complexity)** của thuật toán là thước đo định lượng về tài nguyên (thời gian/bộ nhớ) mà thuật toán tiêu tốn
 - Thường được biểu diễn dưới dạng các hàm của **kích thước đầu vào (input size)**
 - Giúp ta so sánh và lựa chọn thuật toán hiệu quả nhất cho bài toán
- Với các hàm độ phức tạp $f : \mathbb{R} \rightarrow \mathbb{R}$ hoặc $f : \mathbb{N} \rightarrow \mathbb{R}$, ta cần hiểu rõ **tốc độ tăng trưởng** của chúng
 - Ta nói f **tăng nhanh hơn** g nếu với mọi giá trị x đủ lớn, $f(x) > g(x)$
 - Giúp ta đánh giá hiệu quả tương đối giữa các thuật toán khi kích thước đầu vào tăng lên
 - Là nền tảng để phân loại các thuật toán theo hiệu quả tiệm cận
- Để đánh giá và so sánh độ phức tạp một cách chính xác, ta sử dụng **ký hiệu O-lớn (big-O notation)** và các ký hiệu tiệm cận khác
 - Cho phép ta mô tả **tốc độ tăng trưởng (growth rate)** của các hàm độ phức tạp
 - Tập trung vào các thành phần chính ảnh hưởng đến tốc độ tăng trưởng, bỏ qua các yếu tố không quan trọng khi kích thước đầu vào lớn

21

66

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Ký hiệu O -lớn

Cho f và g là các hàm $\mathbb{R} \rightarrow \mathbb{R}$. Ta nói rằng f là $O(g)$ (đọc là “ f là O -lớn của g ” hoặc “ f thuộc lớp $O(g)$ ”) nếu **tồn tại các hằng số C và k sao cho $|f(x)| \leq C|g(x)|$ với mọi $x > k$**

Ký hiệu O -lớn

Cho f và g là các hàm $\mathbb{R} \rightarrow \mathbb{R}$. Giả sử $g(x) \neq 0$ với $x \in \mathbb{R}$ đủ lớn. Ta nói rằng f là $O(g)$ (đọc là “ f là O -lớn của g ” hoặc “ f thuộc lớp $O(g)$ ”) nếu $\lim_{x \rightarrow \infty} \frac{|f(x)|}{|g(x)|}$ **là hữu hạn**

- Nếu f là $O(g)$, ta cũng nói “ f bị chặn trên bởi g ”
- Các hằng số C và k được gọi là các **bằng chứng (witness)** cho mối liên hệ giữa f và g . **Để xác định f là $O(g)$, chỉ cần một cặp bằng chứng là đủ**

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

22

66

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

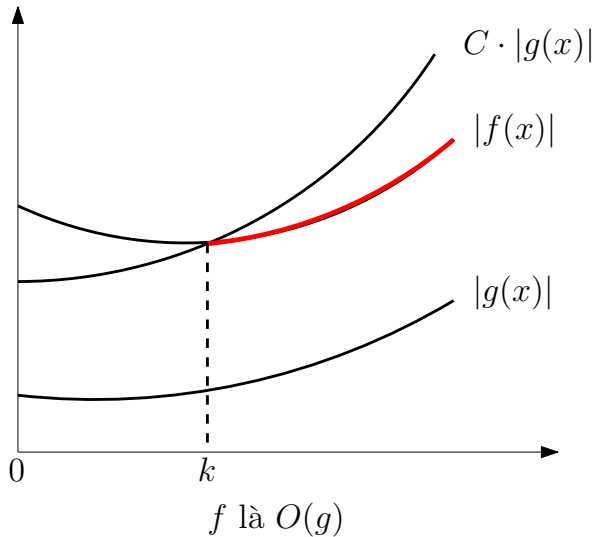
Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn



23

66

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Ví dụ 6

Ta chứng minh hàm $f : \mathbb{R} \rightarrow \mathbb{R}$ cho bởi $f(x) = x^2 + 2x + 1$ là $O(g)$ với $g(x) = x^2$ (Ta cũng viết $x^2 + 2x + 1$ là $O(x^2)$)

Cách 1:

- Chú ý rằng khi $x > 1$, ta có $x < x^2$ và $1 < x^2$
- Do đó với mọi $x > 1$, ta có

$$|f(x)| = |x^2 + 2x + 1| \leq |x^2 + 2x^2 + x^2| = 4|x^2|$$

- Ta chọn $C = 4$ và $k = 1$

Cách 2: Do $\lim_{x \rightarrow \infty} \frac{|f(x)|}{|g(x)|} = \lim_{x \rightarrow \infty} \frac{|x^2 + 2x + 1|}{|x^2|} = 1$, ta có $f = O(g)$

24

66

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Chú ý về quy tắc chia (division law of limit)

$$\lim_{x \rightarrow a} \frac{f(x)}{g(x)} = \frac{\lim_{x \rightarrow a} f(x)}{\lim_{x \rightarrow a} g(x)}$$

chỉ đúng khi $\lim_{x \rightarrow a} f(x)$ và $\lim_{x \rightarrow a} g(x)$ đều tồn tại^a và $\lim_{x \rightarrow a} g(x) \neq 0$

^aNghĩa là các giới hạn này là các số hữu hạn

Nhắc lại: Quy tắc L'Hospital (L'Hospital's rule)

Nếu $\lim_{x \rightarrow \infty} f(x) = \infty$ và $\lim_{x \rightarrow \infty} g(x) = \infty$, thì $\lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = \lim_{x \rightarrow \infty} \frac{f'(x)}{g'(x)}$, trong đó $f'(x)$ và $g'(x)$ lần lượt là đạo hàm của $f(x)$ và $g(x)$

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

25

66

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Ví dụ 7

Ta chứng minh x^2 là $O(2^x)$ bằng cách sử dụng định nghĩa O -lớn theo giới hạn. Ta có

$$\begin{aligned}\lim_{x \rightarrow \infty} \frac{|x^2|}{|2^x|} &= \lim_{x \rightarrow \infty} \frac{x^2}{2^x} \\&= \lim_{x \rightarrow \infty} \frac{2x}{2^x \cdot \ln 2} \\&= \frac{2}{\ln 2} \lim_{x \rightarrow \infty} \frac{x}{2^x} \\&= \frac{2}{\ln 2} \lim_{x \rightarrow \infty} \frac{1}{2^x \cdot \ln 2} \\&= 0.\end{aligned}$$

Do đó, theo định nghĩa, x^2 là $O(2^x)$

Quy tắc L'Hospital

Quy tắc L'Hospital

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

26

66

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Chú ý

Nếu không đề cập gì thêm thì $\log(n) = \log_2(n)$

Bài tập 2

Chứng minh

- (a) $7x$ là $O(x^3)$
- (b) x^3 không là $O(x^2)$
- (c) $1 + 2 + \dots + n$ là $O(n^2)$
- (d) $n! = 1 \times 2 \times \dots \times n$ là $O(n^n)$
- (e) $\log(n!)$ là $O(n \log n)$
- (f) n^3 là $O(2^n)$
- (g) $\log n$ là $O(n)$
- (h) Với các hằng số $b > 1$ và $k > 0$, $\log_b(n^k)$ là $O(\log n)$

Bài tập 3

Hãy giải thích một hàm $f: \mathbb{R} \rightarrow \mathbb{R}$ là $O(1)$ nghĩa là gì

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

27

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

28

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Bài tập 4

Chứng minh rằng

(a) x^3 là $O(x^4)$ nhưng x^4 không là $O(x^3)$

(b) $3x^4 + 1$ là $O(x^4/2)$ và $x^4/2$ là $O(3x^4 + 1)$

(c) $x \log x$ là $O(x^2)$ nhưng x^2 không là $O(x \log x)$

(d) 2^n là $O(3^n)$ nhưng 3^n không là $O(2^n)$

Bài tập 5

Chứng minh rằng nếu $f(x)$ là $O(x)$ thì $f(x)$ cũng là $O(x^2)$

Bài tập 6

Chứng minh hoặc tìm phản ví dụ cho phát biểu: Nếu $f_1(x)$ là $O(g_1(x))$ và $f_2(x)$ là $O(g_2(x))$ thì $f_1(x) - f_2(x)$ là $O(g_1(x) - g_2(x))$

Bài tập 7

Chứng minh rằng nếu $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ với $a_0, a_1, \dots, a_n$ là các số thực (nghĩa là, $f(x)$ là một đa thức bậc n) thì f là $O(x^n)$

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

29

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

- Ký hiệu $f(n) = O(g(n))$ thường được sử dụng để chỉ $f(n)$ là $O(g(n))$
 - Ký hiệu này không hoàn toàn chặt chẽ về mặt toán học, do $f(n)$ là một hàm còn $O(g(n))$ là một tập hợp các hàm
 - $f(n) = O(g(n))$ trên thực tế nghĩa là $f(n) \in O(g(n))$, do đó có thể viết $n = O(n^2)$ nhưng **không nên viết $O(n^2) = n$**
- Bạn có thể gặp biểu thức dạng “ $f(n) + O(g(n)) = O(h(n))$ ”
 - Dấu “=” ở đây nghĩa là “ $\subseteq$ ”. Cụ thể, biểu thức trên cần được hiểu là tập hợp S gồm các hàm $f(n) + g_1(n)$ với $g_1(n) \in O(g(n))$ là tập con của tập $O(h(n))$
- Bạn có thể gặp biểu thức dạng “ $f(n) \leq g(n) + O(h(n))$ với mọi $n \geq 0$ ” hoặc tương tự
 - Nghĩa là tồn tại $e(n)$ sao cho (a) $f(n) \leq g(n) + e(n)$ với mọi $n \geq 0$ và (b) $e(n) \in O(h(n))$
- Một số tác giả định nghĩa O -lớn bằng cách thay điều kiện $|f(x)| \leq C|g(x)|$ bằng $0 \leq f(x) \leq C(g(x))$. (Làm việc với giá trị tuyệt đối và khả năng các hàm $f(x)$ và $g(x)$ có thể nhận giá trị âm thường khó hơn là chỉ làm việc với các hàm nhận giá trị dương.) Định nghĩa theo cách này không hoàn toàn chặt chẽ. Ví dụ như hàm $\log x$ có thể nhận giá trị âm với x nhỏ

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Bài tập 8

Giải thích ý nghĩa của các ký hiệu sau:

(a) $n^{O(k)}$

(b) $n^2 + O(n)$

(c) $n^2 + O(n) = O(n^2)$

Bài tập 9

Sử dụng ký hiệu O -lớn để đánh giá liên hệ giữa các cặp hàm sau. Giải thích kết quả của bạn

(a) n^{124} và 1.24^n

(b) $^{124}\sqrt{n}$ và $(\log n)^{124}$

(c) $n \log n$ và $n^{\log n}$

(d) $\sqrt{n}$ và $2^{\sqrt{\log n}}$

(e) $\sqrt{n}$ và $n^{\sin(n)}$

(f) $(n + \log n)^2$ và $n^2 + n \log n$

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

30

66

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

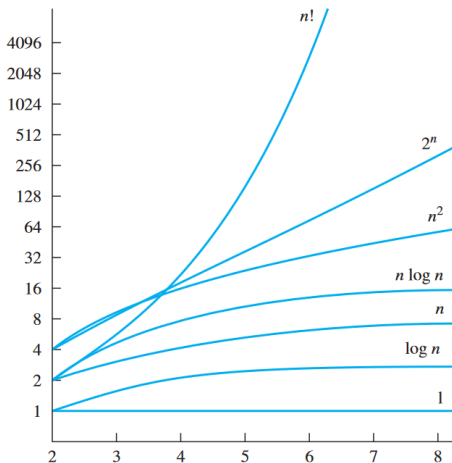
Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn



Hình: Độ tăng của một số hàm thường dùng khi đánh giá với ký hiệu O -lớn [Rosen 2012]

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số xấp xỉ hữu ích

(1) Nếu $d > c > 1$, thì n^c là $O(n^d)$, nhưng n^d không là $O(n^c)$

(2) Nếu $b > 1$ và c, d là các số dương, thì $(\log_b n)^c$ là $O(n^d)$ nhưng n^d không là $O((\log_b n)^c)$

(3) Nếu $b > 1$ và d là số dương, thì n^d là $O(b^n)$ nhưng b^n không là $O(n^d)$

(4) Nếu $c > b > 1$, thì b^n là $O(c^n)$ nhưng c^n không là $O(b^n)$

Một số tính chất quan trọng

(a) Nếu $f_1(x) = O(g_1(x))$ và $f_2(x) = O(g_2(x))$ thì $(f_1 + f_2)(x) = O(g(x))$ trong đó $g(x) = \max\{|g_1(x)|, |g_2(x)|\}$ với mọi $x \in \mathbb{R}$

(b) Nếu $f_1(x) = O(g_1(x))$ và $f_2(x) = O(g_2(x))$ thì $(f_1 f_2)(x) = O(g_1(x)g_2(x))$

Bài tập 10

Ước lượng theo O -lớn hàm $f(n) = 3n \log n! + (n^2 + 3) \log n$

32

66

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

33

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Bài tập 11

Sắp xếp các hàm sau trong một danh sách sao cho mỗi hàm là O -lớn của hàm tiếp theo

- (a) $\sqrt{n}$, $1000 \log n$, $n \log n$, $2n!$, 2^n , 3^n , và $\frac{n^2}{1000000}$
- (b) $(1.5)^n$, n^{100} , $(\log n)^3$, $\sqrt{n} \log n$, 10^n , $(n!)^2$, và $n^{99} + n^{98}$

Bài tập 12

Giả sử bạn có hai thuật toán khác nhau để giải một bài toán. Để giải một bài toán có kích thước n ,

- (1) thuật toán thứ nhất sử dụng đúng $n(\log n)$ phép toán và thuật toán thứ hai sử dụng đúng $n^{3/2}$ phép toán;
- (2) thuật toán thứ nhất sử dụng đúng $n^2 2^n$ phép toán và thuật toán thứ hai sử dụng đúng $n!$ phép toán.

Khi n tăng, trong mỗi trường hợp trên, thuật toán nào sử dụng ít phép toán hơn?

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Bài tập 13

Hãy đưa ra ước lượng O -lớn chính xác nhất có thể cho mỗi hàm sau

- (a) $(n^2 + 8)(n + 1)$
- (b) $(n \log n + n^2)(n^3 + 2)$
- (c) $(n! + 2^n)(n^3 + \log(n^2 + 1))$
- (d) $(n^3 + n^2 \log n)(\log n + 1) + (17 \log n + 19)(n^3 + 2)$
- (e) $(2n + n^2)(n^3 + 3n)$
- (f) $(n^n + n2^n + 5^n)(n! + 5n)$
- (g) $n \log(n^2 + 1) + n^2 \log n$
- (h) $(n \log n + 1)^2 + (\log n + 1)(n^2 + 1)$
- (i) $n^{2^n} + n^{n^2}$

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

34

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Bài tập 14

Giả sử $T(1) = O(1)$ và $T(n) = T(n-1) + O(1)$ với mọi $n > 1$. Chứng minh sau sai ở đâu? Hãy đánh giá $T(n)$ theo ký hiệu O -lớn và giải thích đánh giá của bạn.

Giả sử $P(n)$ là phát biểu $T(n) = O(1)$. Ta chứng minh $\forall n \in \mathbb{Z}^+ P(n)$ bằng quy nạp.

- **Bước cơ sở:** $P(1)$ đúng vì $T(1) = O(1)$.
- **Bước quy nạp:** Giả sử $P(k)$ đúng với số nguyên $k \geq 1$ nào đó, nghĩa là $T(k) = O(1)$. Ta chứng minh $P(k+1)$ đúng, nghĩa là chứng minh $T(k+1) = O(1)$. Thật vậy, ta có:

$$T(k+1) = T(k) + O(1) \quad \text{giả thiết} \quad (1)$$

$$= O(1) + O(1) \quad \text{giả thiết quy nạp} \quad (2)$$

$$= O(1). \quad (3)$$

Do đó, $P(k+1)$ đúng.

Vậy theo nguyên lý quy nạp, ta có $P(n)$ đúng với mọi $n \in \mathbb{Z}^+$, tức là $T(n) = O(1)$ với mọi $n \in \mathbb{Z}^+$.

35

66

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Ký hiệu Ω -lớn

Cho f và g là các hàm $\mathbb{R} \rightarrow \mathbb{R}$. Ta nói rằng f là $\Omega(g)$ nếu tồn tại các hằng số $C > 0$ và k sao cho $|f(x)| \geq C|g(x)|$ với mọi $x > k$

Ký hiệu Ω -lớn

Cho f và g là các hàm $\mathbb{R} \rightarrow \mathbb{R}$. Giả sử $g(x) \neq 0$ với $x \in \mathbb{R}$ đủ lớn. Ta nói rằng f là $\Omega(g)$ nếu $\lim_{x \rightarrow \infty} \frac{|f(x)|}{|g(x)|}$ khác 0

■ Nếu f là $\Omega(g)$, ta cũng nói “ f bị chặn dưới bởi g ”

Bài tập 15

Chứng minh rằng f là $\Omega(g)$ khi và chỉ khi g là $O(f)$

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

36

66

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Chú ý

f là $O(g)$ theo nghĩa nào đó là “độ tăng của $f \leq$ độ tăng của g ” và tương tự f là $\Omega(g)$ nghĩa là “độ tăng của $f \geq$ độ tăng của g ”. Tuy nhiên, **bạn cần cẩn thận!** Với hai số thực $a, b \in \mathbb{R}$, các bất đẳng thức $a \leq b$ và $b \leq a$ không thể cùng sai. Nhưng tồn tại hàm f sao cho $f = O(g)$ và $f = \Omega(g)$ cùng sai

Bài tập 16 (★)

Tìm các ví dụ của các hàm $f : \mathbb{R} \rightarrow \mathbb{R}$ thỏa mãn các điều kiện (a) – (d) tương ứng

	$f(n)$ là $O(n^3)$	$f(n)$ không là $O(n^3)$
$f(n)$ là $\Omega(n^3)$	(a)	(b)
$f(n)$ không là $\Omega(n^3)$	(c)	(d)

Cụ thể, ở (a), bạn cần tìm ví dụ về một hàm $f(n)$ đồng thời là $O(n^3)$ và $\Omega(n^3)$ và chứng minh ví dụ bạn tìm ra là đúng. Tương tự cho các phần (b), (c), và (d)

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

37

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

38

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Ký hiệu Θ -lớn

Cho f và g là các hàm $\mathbb{R} \rightarrow \mathbb{R}$. Ta nói rằng f là $\Theta(g)$ nếu f là $O(g)$ và f là $\Omega(g)$

Ký hiệu Θ -lớn

Cho f và g là các hàm $\mathbb{R} \rightarrow \mathbb{R}$. Giả sử $g(x) \neq 0$ với $x \in \mathbb{R}$ đủ lớn. Ta nói rằng f là $\Theta(g)$ nếu $\lim_{x \rightarrow \infty} \frac{|f(x)|}{|g(x)|}$ là hữu hạn và khác 0

Bài tập 17

- (a) Chứng minh rằng $1 + 2 + \dots + n$ là $\Theta(n^2)$
- (b) Các tập $O(1)$ và $\Theta(1)$ có bằng nhau không? Tại sao?

Bài tập 18

Chứng minh rằng với các hàm f, g từ $\mathbb{R}$ đến $\mathbb{R}$, f là $\Theta(g)$ khi và chỉ khi tồn tại các hằng số dương C_1, C_2 , và k sao cho $C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$ với mọi $x > k$

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Ký hiệu o -nhỏ

Cho f và g là các hàm $\mathbb{R} \rightarrow \mathbb{R}$. Ta nói rằng f là $o(g)$ nếu với mọi $C > 0$ tồn tại k sao cho $|f(x)| < C|g(x)|$ với mọi $x > k$

Ký hiệu o -nhỏ

Cho f và g là các hàm $\mathbb{R} \rightarrow \mathbb{R}$. Giả sử $g(x) \neq 0$ với $x \in \mathbb{R}$ đủ lớn. Ta nói rằng f là $o(g)$ nếu $\lim_{x \rightarrow \infty} \frac{|f(x)|}{|g(x)|} = 0$

Ký hiệu o -nhỏ

Cho f và g là các hàm $\mathbb{R} \rightarrow \mathbb{R}$. Ta nói rằng f là $o(g)$ nếu f là $O(g)$ nhưng f không là $\Omega(g)$

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

39

66

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Ký hiệu ω -nhỏ

Cho f và g là các hàm $\mathbb{R} \rightarrow \mathbb{R}$. Ta nói rằng f là $\omega(g)$ nếu với mọi $C > 0$ tồn tại k sao cho $|f(x)| > C|g(x)|$ với mọi $x > k$

Ký hiệu ω -nhỏ

Cho f và g là các hàm $\mathbb{R} \rightarrow \mathbb{R}$. Giả sử $g(x) \neq 0$ với $x \in \mathbb{R}$ đủ lớn. Ta nói rằng f là $\omega(g)$ nếu $\lim_{x \rightarrow \infty} \frac{|f(x)|}{|g(x)|}$ bằng ∞

Ký hiệu ω -nhỏ

Cho f và g là các hàm $\mathbb{R} \rightarrow \mathbb{R}$. Giả sử $g(x) \neq 0$ với $x \in \mathbb{R}$ đủ lớn. Ta nói rằng f là $\omega(g)$ nếu f là $\Omega(g)$ nhưng f không là $O(g)$

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

40

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Độ phức tạp tính toán

Độ tăng của các hàm



Thuật toán I

Hoàng Anh Đức

Tóm lại, với các hàm f và g từ $\mathbb{R}$ đến $\mathbb{R}$

■ f là $O(g)$ “ f tăng không nhanh hơn g ” tương tự “ $\leq$ ”

$$\exists C, k \forall x > k \quad |f(x)| \leq C|g(x)|$$

■ f là $\Omega(g)$ “ f tăng ít nhất nhanh như g ” tương tự “ $\geq$ ”

$$\exists C > 0, k \forall x > k \quad |f(x)| \geq C|g(x)|$$

■ f là $\Theta(g)$ “ f tăng nhanh như g ” tương tự “ $=$ ”

$$\exists C_1 > 0, C_2 > 0, k \forall x > k \quad C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$$

■ f là $o(g)$ “ f tăng chậm hơn g ” tương tự “ $<$ ”

$$\forall C > 0 \exists k \forall x > k \quad |f(x)| < C|g(x)|$$

■ f là $\omega(g)$ “ f tăng nhanh hơn g ” tương tự “ $>$ ”

$$\forall C > 0 \exists k \forall x > k \quad |f(x)| > C|g(x)|$$

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

41

66

Độ phức tạp tính toán

Định nghĩa và khái niệm



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

- Một thuật toán tốt phải *luôn đúng về mặt kết quả* và *hiệu quả về mặt tài nguyên*
- **Độ phức tạp (complexity)** đo lường mức độ khó khăn của một tính toán dựa trên tài nguyên cần thiết:
 - **Độ phức tạp theo thời gian (time complexity)**: Số lượng các phép toán cơ bản hoặc số bước thực hiện
 - **Độ phức tạp theo không gian (space complexity)**: Lượng bộ nhớ (số bit) cần thiết để thực hiện tính toán
- Độ phức tạp của thuật toán thay đổi theo **kích thước đầu vào (input size)**
 - Ví dụ: Tìm kiếm một phần tử trong dãy có 100000 phần tử sẽ tốn nhiều tài nguyên hơn so với dãy có chỉ vài phần tử
- Do đó, độ phức tạp thường được biểu diễn dưới dạng một **hàm số $f(n)$** với n là kích thước đầu vào
 - Hàm này giúp ta dự đoán tài nguyên cần thiết khi kích thước đầu vào tăng lên
- Khi phân tích thuật toán, chúng ta không quan tâm đến chính xác số lượng phép toán, mà chỉ cần **đánh giá tiệm cận (asymptotic estimate)** về mức độ tài nguyên tiêu thụ khi kích thước đầu vào tăng lên

42

66

Độ phức tạp tính toán

Độ phức tạp tính toán theo thời gian



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

43

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

- Các ký hiệu tiệm cận là những công cụ toán học thiết yếu để đặc trưng hóa và so sánh độ phức tạp các thuật toán
- Trong phân tích *độ phức tạp tính toán theo thời gian (time complexity)*, ta xem xét ba loại trường hợp:
 - *Độ phức tạp trường hợp xấu nhất (worst-case complexity)*: Thời gian tối đa mà thuật toán có thể tiêu tốn với bất kỳ đầu vào nào có kích thước cụ thể—**thường được sử dụng phổ biến nhất trong phân tích thuật toán**
 - *Độ phức tạp trường hợp trung bình (average-case complexity)*: Thời gian trung bình khi xét tất cả các đầu vào có khả năng xuất hiện với cùng kích thước—**thường khó xác định**
 - *Độ phức tạp trường hợp tốt nhất (best-case complexity)*: Thời gian tối thiểu với đầu vào thuận lợi nhất có kích thước cụ thể—**ít có giá trị thực tiễn trong đánh giá thuật toán**
- Trong thực hành, ta thường mô tả thời gian chạy của thuật toán theo *ký hiệu O-lớn* để đơn giản hóa việc phân tích. *Ký hiệu Θ -lớn* cung cấp đánh giá chính xác hơn về độ phức tạp thực sự nhưng đôi khi khó xác định
- Cần lưu ý rằng nhiều tài liệu sử dụng biểu thức “ $f(n)$ là $O(g(n))$ ” khi thực chất họ đang muốn biểu diễn “ $f(n)$ là $\Theta(g(n))$ ”

Độ phức tạp tính toán

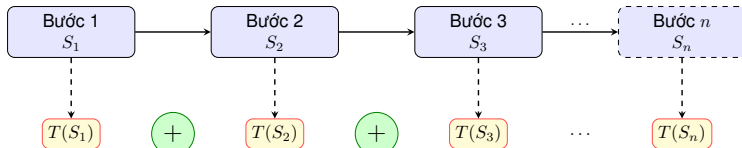
Độ phức tạp tính toán theo thời gian



Thuật toán I

Hoàng Anh Đức

Độ phức tạp của một chuỗi tuần tự các bước là tổng của độ phức tạp của từng bước



$$\text{Tổng độ phức tạp} = T(S_1) + T(S_2) + T(S_3) + \dots + T(S_n)$$

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

44

66

Độ phức tạp tính toán

Độ phức tạp tính toán theo thời gian



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

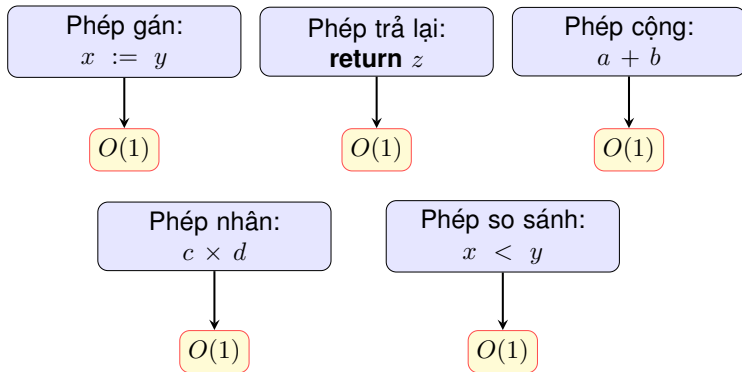
Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Thời gian thực hiện các lệnh gán ($:=$), trả lại (**return**) là $O(1)$, và giả sử thời gian thực hiện các phép toán cơ bản (cộng, trừ, nhân, chia, so sánh, v.v...) cũng là $O(1)$

- Trong thực tế, việc cộng hai số nguyên độ dài n bit biểu diễn dưới dạng số nhị phân có độ phức tạp $O(n)$ chứ không phải $O(1)$



45

66

Độ phức tạp tính toán

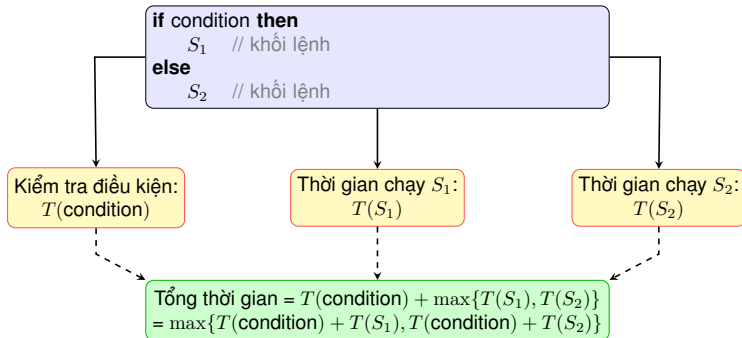
Độ phức tạp tính toán theo thời gian



Thuật toán I

Hoàng Anh Đức

Thời gian thực hiện cấu trúc **if...then...else** là thời gian lớn nhất thực hiện các lệnh sau **then** hoặc sau **else** cộng với thời gian kiểm tra điều kiện



Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

46

66

Độ phức tạp tính toán

Độ phức tạp tính toán theo thời gian



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

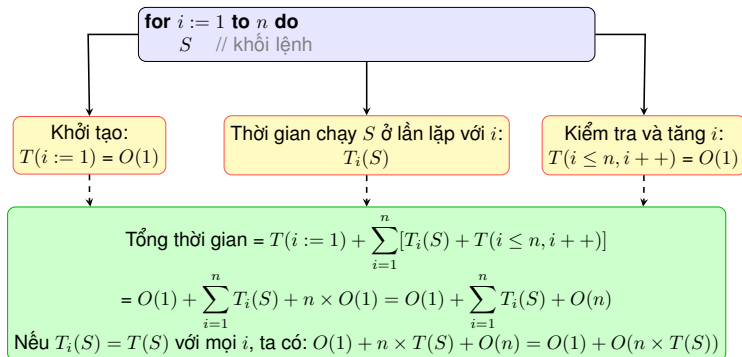
Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Thời gian thực hiện vòng lặp là tổng thời gian thực hiện các lần lặp và thời gian kiểm tra điều kiện lặp. Nếu thời gian thực hiện mỗi lần lặp là giống nhau, thì tổng thời gian thực hiện các lần lặp là tích của số lần lặp và thời gian thực hiện mỗi lần lặp



47

66

Độ phức tạp tính toán

Độ phức tạp tính toán theo thời gian



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

48

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

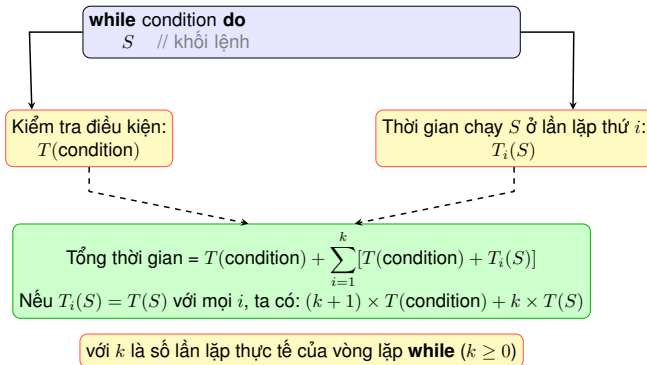
Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Thời gian thực hiện vòng lặp là tổng thời gian thực hiện các lần lặp và thời gian kiểm tra điều kiện lặp. Nếu thời gian thực hiện mỗi lần lặp là giống nhau, thì tổng thời gian thực hiện các lần lặp là tích của số lần lặp và thời gian thực hiện mỗi lần lặp



Chú ý: Nếu điều kiện sai ngay từ đầu ($k = 0$), khối lệnh S không được thực hiện lần nào

Độ phức tạp tính toán

Độ phức tạp tính toán theo thời gian



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

49 Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

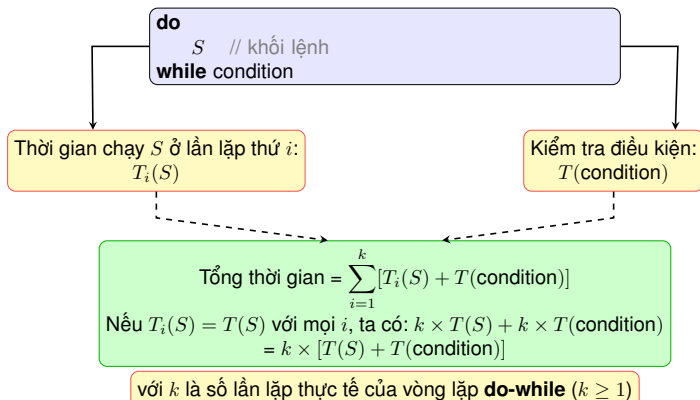
Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Thời gian thực hiện vòng lặp là tổng thời gian thực hiện các lần lặp và thời gian kiểm tra điều kiện lặp. Nếu thời gian thực hiện mỗi lần lặp là giống nhau, thì tổng thời gian thực hiện các lần lặp là tích của số lần lặp và thời gian thực hiện mỗi lần lặp



Chú ý: Khác vòng lặp **while**, vòng lặp **do-while** luôn thực hiện S ít nhất một lần

Độ phức tạp tính toán

Độ phức tạp tính toán theo thời gian



Thuật toán I

Hoàng Anh Đức

Bảng: Một số thuật ngữ thường dùng

Độ phức tạp	Thuật ngữ
$O(1)$	Độ phức tạp hằng số (constant complexity)
$O(\log n)$	Độ phức tạp lôgarit (logarithmic complexity)
$O(n)$	Độ phức tạp tuyến tính (linear complexity)
$O(n \log n)$	Độ phức tạp $n \log n$ (linearithmic complexity)
$O(n^b)$	Độ phức tạp đa thức (polynomial complexity)
$O(b^n)$, với $b > 1$	Độ phức tạp hàm mũ (exponential complexity)
$O(n!)$	Độ phức tạp giai thừa (factorial complexity)

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

50

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Độ phức tạp tính toán

Độ phức tạp tính toán theo thời gian



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Thuật toán 1: Tìm giá trị của phần tử lớn nhất

Input: $a_1, a_2, \dots, a_n$: dãy số nguyên

Output: Giá trị của phần tử lớn nhất trong dãy

```
1  $v := a_1$   $t_1$ 
2 for  $i := 2$  to  $n$  do
3   if  $a_i > v$  then  $t_2$ 
4      $v := a_i$   $t_4$ 
5 return  $v$   $t_3$ 
```

$$T(n) = t_1 + t_2 + t_3$$

$$t_1, t_3 \text{ là } O(1)$$

$T(n)$ là $O(n)$

(xấu nhất)

$T(n)$ là $O(n)$

(tốt nhất)

$$t_2 = O(1) + \sum_{i=2}^n t_4^i + O(n)$$

$$t_4^i = t_5 + (\text{t.g. kiểm tra } a_i > v) = O(1) + O(1) = O(1)$$

51

66

Thuật toán

Một số bài tập



Bài tập 19

- (a) Thiết kế thuật toán để tính tổng của tất cả các số hạng trong một dãy số nguyên $a_1, a_2, \dots, a_n$ cho trước
- (b) Chứng minh thuật toán bạn thiết kế là đúng
- (c) Đánh giá thời gian chạy của thuật toán bạn thiết kế

Bài tập 20

Một chuỗi ký tự được gọi là **chuỗi đối xứng (palindrome)** khi viết từ trái qua phải và viết từ phải qua trái thì chuỗi không thay đổi. Một ví dụ là chuỗi *madam*.

- (a) Thiết kế thuật toán để kiểm tra xem một chuỗi ký tự có phải là chuỗi đối xứng hay không
- (b) Đánh giá thời gian chạy của thuật toán bạn thiết kế

Bài tập 21

Cho $f: A \rightarrow B$ là một hàm với các tập A, B là các tập con hữu hạn của $\mathbb{Z}$. Hãy thiết kế một thuật toán để kiểm tra xem

- (a) liệu f có là đơn ánh không;
- (b) liệu f có là toàn ánh không.

Trong mỗi trường hợp, hãy đánh giá thời gian chạy của thuật toán bạn thiết kế

Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

52

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Thuật toán

Một số thuật toán tìm kiếm



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

53

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Bài toán tìm kiếm

Cho một dãy n phần tử $a_1, a_2, \dots, a_n$ và một phần tử x . Tìm x trong dãy đã cho hoặc kết luận rằng x không có trong dãy

- Tìm kiếm tuyến tính (Linear Search)
- Tìm kiếm nhị phân (Binary Search)

66



Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

■ Bài toán:

- **Input:** $a_1, \dots, a_n$: dãy số nguyên, x : số nguyên
- **Output:** Chỉ số i thỏa mãn $x = a_i$ hoặc 0 nếu x không có trong dãy

■ Tìm kiếm tuyến tính:

- (1) Bắt đầu từ phần tử đầu tiên của dãy (a_1)
- (2) So sánh phần tử hiện tại (a_i) với giá trị cần tìm x
- (3) Nếu $a_i = x$, trả về vị trí i và kết thúc thuật toán
- (4) Nếu $a_i \neq x$, chuyển sang phần tử tiếp theo (a_{i+1})
- (5) Lặp lại bước (2) đến (4) cho đến khi tìm thấy x hoặc đã duyệt hết dãy
- (6) Nếu đã duyệt hết dãy mà không tìm thấy x , trả về giá trị 0



Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Ví dụ 8 (Tìm kiếm tuyến tính)

- **Input:** Dãy $a_1 = 2, a_2 = 5, a_3 = 6, a_4 = 8, a_5 = 12$ và $x = 8$
- **Output:** Chỉ số i thỏa mãn $x = a_i$ hoặc 0 nếu x không có trong dãy

a_1	a_2	a_3	a_4	a_5
2	5	6	8	12
$i = 1$	$i = 2$	$i = 3$	$i = 4$	
$\neq x$	$\neq x$	$\neq x$	$= x$	



Thuật toán 3: Tìm kiếm tuyến tính (Linear Search)

Input: $a_1, \dots, a_n$: dãy số nguyên, x : số nguyên

Output: Chỉ số i thỏa mãn $x = a_i$ hoặc 0 nếu x không có trong dãy

```
1  $i := 1$  // Bắt đầu từ đầu dãy
2 while  $i \leq n$  và  $x \neq a_i$  do // Chưa xong và chưa tìm thấy
3    $i := i + 1$  // Đi tới vị trí tiếp theo trong dãy
4 if  $i \leq n$  then
5    $location := i$  // Tìm thấy  $x$  trong dãy
6 else
7    $location := 0$  // Không tìm thấy  $x$  trong dãy
8 return  $location$ 
```

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn



Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

■ Bài toán:

- **Input:** $a_1, \dots, a_n$: dãy số nguyên *thực sự tăng*, x : số nguyên
- **Output:** Chỉ số i thỏa mãn $x = a_i$ hoặc 0 nếu x không có trong dãy

■ Tìm kiếm nhị phân: (Một ví dụ về kỹ thuật *chia để trị* (*divide and conquer*) trong thiết kế thuật toán)

- (1) Tính $m = \lfloor (1 + n)/2 \rfloor$. Phần tử ở giữa của dãy là a_m
- (2) Chia dãy $a_1, \dots, a_n$ thành hai dãy con (a) $a_1, \dots, a_m$ và (b) $a_{m+1}, \dots, a_n$. Nếu $x > a_m$ thì ta chỉ tìm x trong dãy con (b), còn ngược lại thì ta chỉ tìm x trong dãy con (a)
- (3) Làm tương tự cho đến khi không gian tìm kiếm chỉ còn một phần tử a_i . Nếu $x = a_i$ thì trả lại vị trí i của x , còn ngược lại thì trả lại 0



Ví dụ 9 (Tìm kiếm nhị phân)

- Input:** Dãy $a_1 = 2, a_2 = 5, a_3 = 6, a_4 = 8, a_5 = 12$ và $x = 8$
- Output:** Chỉ số i thỏa mãn $x = a_i$ hoặc 0 nếu x không có trong dãy

a_1	a_2	a_3	a_4	a_5
2	5	6	8	12
i		m		j

$< x$

a_1	a_2	a_3	a_4	a_5
2	5	6	8	12
			$i = m$	j
			$= x$	

a_1	a_2	a_3	a_4	a_5
2	5	6	8	12
			$i = j$	
			$= x$	

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn



Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Thuật toán 4: Tìm kiếm nhị phân (Binary Search)

Input: $a_1, \dots, a_n$: dãy số nguyên *thực sự tăng*, x : số nguyên

Output: Chỉ số i thỏa mãn $x = a_i$ hoặc 0 nếu x không có trong dãy

```
1   $i := 1$                                 // Chỉ số bắt đầu khoảng tìm kiếm
2   $j := n$                                 // Chỉ số kết thúc khoảng tìm kiếm
3  while  $i < j$  do                        // Khi khoảng tìm kiếm có  $> 1$  phần tử
4       $m := \lfloor (i + j) / 2 \rfloor$             // Chỉ số của phần tử ở giữa
5      if  $x > a_m$  then
6           $i := m + 1$ 
7      else
8           $j := m$ 
9  if  $x = a_i$  then
10      $location := i$ 
11 else
12      $location := 0$ 
13 return  $location$ 
```

Thuật toán

Một số thuật toán sắp xếp



Thuật toán I

Hoàng Anh Đức

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

60

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Bài toán sắp xếp

Cho một dãy n phần tử và một cách so sánh hai phần tử bất kỳ trong dãy. Hãy sắp xếp dãy theo thứ tự tăng dần

- Sắp xếp nổi bọt (Bubble Sort)
- Sắp xếp chèn (Insertion Sort)



Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

■ Bài toán:

- **Input:** $a_1, a_2, \dots, a_n$: dãy số thực ($n \geq 2$)
- **Output:** Dãy đã cho được sắp xếp theo thứ tự tăng dần

■ Sắp xếp nổi bọt:

- (1) So sánh các phần tử liên tiếp, bắt đầu với cặp (a_1, a_2)
- (2) Nếu $a_1 > a_2$, hoán đổi giá trị của chúng
- (3) Lặp lại (1) và (2) với các cặp $(a_2, a_3), (a_3, a_4), \dots, (a_{n-1}, a_n)$. Lúc này, a_n là phần tử lớn nhất trong dãy
- (4) Lặp lại (1) – (3) với dãy $a_1, \dots, a_{n-1}$, và sau đó với dãy $a_1, \dots, a_{n-2}$, dãy $a_1, \dots, a_{n-3}, \dots$, cho đến dãy a_1, a_2

Thuật toán

Sắp xếp nổi bọt



Thuật toán I
Hoàng Anh Đức

Ví dụ 10

- **Input:** Dãy $a_1 = 34, a_2 = 13, a_3 = 21, a_4 = 3, a_5 = 89$
- **Output:** Dãy sắp xếp theo thứ tự tăng dần

	a_1	a_2	a_3	a_4	a_5
	34	13	21	3	89
$i = 1$	13	21	3	34	89
$i = 2$	13	3	21	34	
$i = 3$	3	13	21		
$i = 4$	3	13			
	3	13	21	34	89

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn



Thuật toán 5: Sắp xếp nổi bọt (Bubble Sort)

Input: $a_1, a_2, \dots, a_n$: dãy số thực ($n \geq 2$)

Output: Dãy đã cho được sắp xếp theo thứ tự tăng dần

```
1 for  $i := 1$  to  $n - 1$  do // Lặp lại  $n - 1$  lần
2   for  $j := 1$  to  $n - i$  do
3     if  $a_j > a_{j+1}$  then
4       Hoán đổi giá trị của  $a_j$  và  $a_{j+1}$ 
5   //  $a_{n-i+1}, \dots, a_n$  đã được sắp xếp
6 //  $a_1, \dots, a_n$  đã được sắp xếp
```

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn



Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bất biến vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

■ Bài toán:

- **Input:** $a_1, a_2, \dots, a_n$: dãy số thực ($n \geq 2$)
- **Output:** Dãy đã cho được sắp xếp theo thứ tự tăng dần

■ Sắp xếp chèn:

- (1) Ban đầu, ta xem a_1 như một dãy đã được sắp xếp
- (2) Để xử lý phần tử a_i ($i = 2, 3, \dots, n$):
 - Lưu giá trị a_i vào một biến tạm m
 - Duyệt các phần tử $a_{i-1}, a_{i-2}, \dots, a_1$ từ phải sang trái
 - Đẩy mỗi phần tử lớn hơn m sang phải một vị trí
 - Chèn m vào vị trí thích hợp (ngay sau phần tử đầu tiên nhỏ hơn hoặc bằng m)
- (3) Sau mỗi bước, dãy $a_1, a_2, \dots, a_i$ đã được sắp xếp theo thứ tự tăng dần
- (4) Lặp lại quá trình cho đến khi xử lý xong a_n



- **Input:** Dãy $a_1 = 34, a_2 = 13, a_3 = 21, a_4 = 3, a_5 = 89$
- **Output:** Dãy sắp xếp theo thứ tự tăng dần

	a_1	a_2	a_3	a_4	a_5
	34	13	21	3	89
$i = 2$	13	34	21	3	89
$i = 3$	13	21	34	3	89
$i = 4$	3	13	21	34	89
$i = 5$	3	13	21	34	89

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn



Thuật toán 6: Sắp xếp chèn (Insertion Sort)

Input: $a_1, a_2, \dots, a_n$: dãy số thực ($n \geq 2$)

Output: Dãy đã cho được sắp xếp theo thứ tự tăng dần

```
1 for  $i = 2$  to  $n$  do
2      $m := a_i$  //  $m$  sắp được chèn vào dãy  $a_1, \dots, a_{i-1}$ 
3      $j := i - 1$ 
4     while  $j \geq 1$  và  $m < a_j$  do // Nếu  $m < a_j$ , đẩy  $a_j$ 
        sang phải để có chỗ chèn  $m$ 
5          $a_{j+1} := a_j$ 
6          $j := j - 1$ 
7      $a_{j+1} := m$  // Chèn  $m$ 
8     // Dãy  $a_1, \dots, a_i$  đã được sắp thứ tự
```

Định nghĩa và một số khái niệm

Mô tả thuật toán

Mô tả bằng ngôn ngữ thông thường

Mô tả bằng giả mã

Mô tả bằng sơ đồ khối

Chứng minh thuật toán

Bắt biên vòng lặp

Độ phức tạp tính toán

Độ tăng của các hàm

Định nghĩa và khái niệm

Độ phức tạp tính toán theo thời gian

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Part I

Phụ lục

Nội dung



Thuật toán I

Hoàng Anh Đức

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp

Tài liệu tham khảo

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp



Thuật toán 7: Tìm kiếm tuyến tính (Linear Search)

Input: $a_1, \dots, a_n$: dãy số nguyên, x : số nguyên

Output: Chỉ số i thỏa mãn $x = a_i$ hoặc 0 nếu x không có trong dãy

```
1  $i := 1$  // Bắt đầu từ đầu dãy
2 while  $i \leq n$  và  $x \neq a_i$  do // Chưa xong và chưa tìm thấy
3    $i := i + 1$  // Đi tới vị trí tiếp theo trong dãy
4 if  $i \leq n$  then
5    $location := i$  // Tìm thấy  $x$  trong dãy
6 else
7    $location := 0$  // Không tìm thấy  $x$  trong dãy
8 return  $location$ 
```

Tìm kiếm và Sắp xếp

2

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp

Tài liệu tham khảo

Một bất biến vòng lặp trong Thuật toán 7 (vòng **while** ở Dòng 2–3) tìm kiếm tuyến tính số nguyên x trong dãy $a_1, \dots, a_n$

$$L = \text{“Ở trước lần lặp với biến } i, x \notin \{a_1, \dots, a_{i-1}\}”$$

- **Khởi động** ($i = 1$): Do $i - 1 = 0$, tập $\{a_1, \dots, a_{i-1}\}$ là tập rỗng, và do đó $x \notin \{a_1, \dots, a_{i-1}\}$, nghĩa là L đúng
- **Duy trì**: Giả sử L đúng ở trước lần lặp với $i = k$ nào đó, nghĩa là $x \notin \{a_1, \dots, a_{k-1}\}$. Ta chứng minh L đúng ở trước lần lặp với $i = k + 1$, nghĩa là $x \notin \{a_1, \dots, a_k\}$. Thật vậy, để thực hiện lần lặp $i = k$, điều kiện ở vòng **while** cần được thỏa mãn, nghĩa là $k \leq n$ và $x \neq a_k$. Kết hợp với giả thiết, ta có điều cần chứng minh
- **Dừng**: Vòng lặp **while** kết thúc khi $i = n + 1$ hoặc $x = a_i$ với $1 \leq i \leq n$. Với trường hợp đầu tiên, bất biến vòng lặp L cho ta $x \notin \{a_1, \dots, a_n\}$ và do đó kết luận không tìm được x . Với trường hợp thứ hai, x hiển nhiên thuộc dãy đã cho

Tìm kiếm và Sắp xếp

3

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp

Tài liệu tham khảo

Thuật toán

Tìm kiếm tuyến tính



Thuật toán 3: Tìm kiếm tuyến tính (Linear Search)

Input: $a_1, \dots, a_n$: dãy số nguyên, x : số nguyên

Output: Chỉ số i thỏa mãn $x = a_i$ hoặc 0 nếu x không có trong dãy

```
1  $i := 1$   $t_1$ 
2 while  $i \leq n$  và  $x \neq a_i$  do
3    $i := i + 1$   $t_5^i$   $t_2$ 
4 if  $i \leq n$  then
5    $location := i$   $t_6$ 
6 else
7    $location := 0$   $t_7$ 
8 return  $location$   $t_4$ 
```

$$T(n) = t_1 + t_2 + t_3 + t_4$$

$T(n)$ là $O(n)$

$T(n)$ là $O(1)$

t_1, t_4 là $O(1)$

(xấu nhất)

(tốt nhất)

$$t_3 = \max\{t_6, t_7\} + (\text{thời gian kiểm tra } i \leq n) = O(1)$$

$$t_2 = \sum_{\{i | i \leq n \wedge x \neq a_i\}} \left[t_5^i + (\text{t.g. kiểm tra } i \leq n \text{ và } x \neq a_i) \right]$$

Thuật toán 1

Hoàng Anh Đức

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp

Tài liệu tham khảo



Thuật toán 8: Tìm kiếm nhị phân (Binary Search)

Input: $a_1, \dots, a_n$: dãy số nguyên *thực sự tăng*, x : số nguyên

Output: Chỉ số i thỏa mãn $x = a_i$ hoặc 0 nếu x không có trong dãy

```
1  $i := 1$  // Chỉ số bắt đầu khoảng tìm kiếm
2  $j := n$  // Chỉ số kết thúc khoảng tìm kiếm
3 while  $i < j$  do // Khi khoảng tìm kiếm có  $> 1$  phần tử
4      $m := \lfloor (i + j) / 2 \rfloor$  // Chỉ số của phần tử ở giữa
5     if  $x > a_m$  then
6          $i := m + 1$ 
7     else
8          $j := m$ 
9 if  $x = a_i$  then
10      $location := i$ 
11 else
12      $location := 0$ 
13 return  $location$ 
```

Thuật toán

Tìm kiếm nhị phân



Thuật toán I

Hoàng Anh Đức

Một bất biến vòng lặp trong Thuật toán 8 (vòng **while** ở Dòng 3–8) tìm kiếm nhị phân số nguyên x trong dãy thực sự tăng $a_1, \dots, a_n$

$L :=$ Ở trước mỗi lần lặp với các biến i, j , nếu $x \in \{a_1, \dots, a_n\}$ thì
 $x \in \{a_i, a_{i+1}, \dots, a_j\}$

- **Khởi động** ($i = 1, j = n$): L hiển nhiên đúng
- **Duy trì**: Giả sử ở trước lần lặp với các biến k, ℓ , nếu $x \in \{a_1, \dots, a_n\}$ thì $x \in \{a_k, a_{k+1}, \dots, a_\ell\}$. Ta chứng minh rằng ở trước lần lặp kế tiếp với các biến (a) $k, \lfloor (k + \ell)/2 \rfloor$ hoặc (b) $\lfloor (k + \ell)/2 \rfloor + 1, \ell$, nếu $x \in \{a_1, \dots, a_n\}$ thì tương ứng (a') $x \in \{a_k, \dots, a_{\lfloor (k+\ell)/2 \rfloor}\}$ hoặc (b') $x \in \{a_{\lfloor (k+\ell)/2 \rfloor + 1}, \dots, a_\ell\}$. Với (a), điều kiện ở Dòng 7 cần được thỏa mãn, nghĩa là $x \leq a_{\lfloor (k+\ell)/2 \rfloor}$. Do đó, nếu $x \in \{a_k, a_{k+1}, \dots, a_\ell\}$ thì (a') đúng. Với (b), điều kiện ở Dòng 5 cần được thỏa mãn, nghĩa là $x > a_{\lfloor (k+\ell)/2 \rfloor}$. Do đó, nếu $x \in \{a_k, a_{k+1}, \dots, a_\ell\}$ thì (b') đúng
- **Dừng**: Vòng lặp **while** dừng khi $i = j$, và từ L , ta có nếu $x \in \{a_1, \dots, a_n\}$ thì $x \in \{a_i\}$. Do đó Thuật toán 8 trả lại vị trí chính xác của x hoặc kết luận không tìm được x (Dòng 9–13)

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp

Tài liệu tham khảo

6

Độ phức tạp tính toán

Tìm kiếm nhị phân



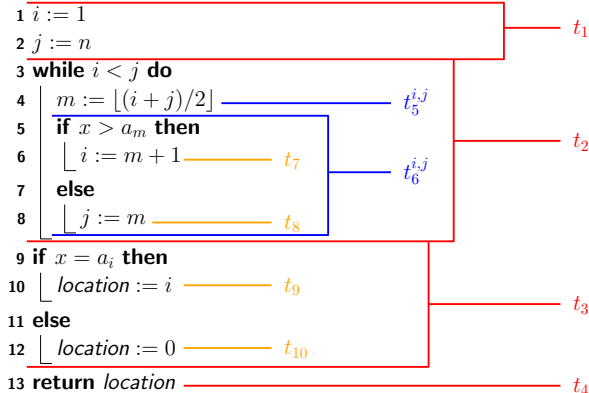
Thuật toán I

Hoàng Anh Đức

Thuật toán 4: Tìm kiếm nhị phân (Binary Search)

Input: $a_1, \dots, a_n$: dãy số nguyên thực sự tăng, x : số nguyên

Output: Chỉ số i thỏa mãn $x = a_i$ hoặc 0 nếu x không có trong dãy



Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp

Tài liệu tham khảo

Độ phức tạp tính toán

Tìm kiếm nhị phân



Thuật toán I

Hoàng Anh Đức

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp

Tài liệu tham khảo

8

$$T(n) = t_1 + t_2 + t_3 + t_4$$

$t_1, t_4, t_5^{i,j}, t_7, \dots, t_{10}$ là $O(1)$

$$t_2 = \sum_{\text{lần lặp while với cặp } i, j} \left[(t_5^{i,j} + t_6^{i,j}) + \right. \\ \left. + (\text{thời gian kiểm tra } i < j) \right]$$

$$t_6^{i,j} = \max\{t_7, t_8\} \\ + (\text{thời gian kiểm tra } x > a_m)$$

$$t_3 = \max\{t_9, t_{10}\} \\ + (\text{thời gian kiểm tra } x = a_i)$$

cặp i, j trong vòng **while** là $O(\log n)$

$$T(n) \text{ là } O(\log n)$$

(xấu nhất)

$$T(n) \text{ là } O(\log n)$$

(tốt nhất)

Độ phức tạp tính toán

Tìm kiếm nhị phân



Thuật toán I

Hoàng Anh Đức

*Tại sao # cặp i, j trong vòng lặp **while** là $O(\log n)$?*

- Lần lặp 0: dãy $a_i, \dots, a_j$ có độ dài $\leq \lceil n/2^0 \rceil$
- Lần lặp 1: dãy $a_i, \dots, a_j$ có độ dài $\leq \lceil n/2^1 \rceil$
- ...
- Lần lặp h (cuối cùng): dãy $a_i, \dots, a_j$ có độ dài $\leq \lceil n/2^h \rceil$
- # cặp i, j trong vòng lặp **while** là $h + 1$
- Vòng **while** kết thúc khi $i = j$, nghĩa là khi dãy $a_i, \dots, a_j$ có độ dài 1
- Vì h là lần lặp cuối cùng, h phải là số nguyên dương nhỏ nhất thỏa mãn $\lceil n/2^h \rceil \leq 1$ (Nếu có số $h' < h$ thỏa mãn $\lceil n/2^{h'} \rceil \leq 1$ thì không thể có lần lặp nào sau h' , trái với giả thiết h là lần lặp cuối cùng)
- Từ $\lceil n/2^h \rceil \leq 1$, suy ra $n/2^h \leq 1$, nghĩa là $h \geq \log n$. Giá trị nhỏ nhất của h là $\lceil \log n \rceil$. Do đó, # cặp i, j trong vòng lặp **while** là $h + 1 = \lceil \log n \rceil + 1 = O(\log n)$

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp

Tài liệu tham khảo

9



Thuật toán 9: Sắp xếp nổi bọt (Bubble Sort)

Input: $a_1, a_2, \dots, a_n$: dãy số thực ($n \geq 2$)

Output: Dãy đã cho được sắp xếp theo thứ tự tăng dần

```
1 for  $i := 1$  to  $n - 1$  do // Lặp lại  $n - 1$  lần
2   for  $j := 1$  to  $n - i$  do
3     if  $a_j > a_{j+1}$  then
4       Hoán đổi giá trị của  $a_j$  và  $a_{j+1}$ 
5   //  $a_{n-i+1}, \dots, a_n$  đã được sắp xếp
6 //  $a_1, \dots, a_n$  đã được sắp xếp
```



Thuật toán 9 có hai vòng lặp **for**: vòng lặp trong ở Dòng 2–4 và vòng lặp ngoài (chứa vòng lặp trong) ở Dòng 1–4

- Một bất biến vòng lặp cho vòng lặp ngoài là

Ở trước lần lặp i , dãy $a_{n-i+1}, \dots, a_n$ là dãy tăng chứa các phần tử lớn hơn hoặc bằng mọi phần tử trong $a_1, \dots, a_{n-i}$

- Một bất biến vòng lặp cho vòng lặp trong là

Ở trước lần lặp j , $a_j = \max\{a_1, \dots, a_j\}$

Sơ đồ chứng minh:

- Chứng minh bước **Khởi động** cho bất biến vòng lặp ngoài
- Ở bước **Duy trì** cho vòng lặp ngoài
 - Chứng minh bất biến vòng lặp trong (**Khởi động**, **Duy trì**, **Dừng**)
 - Sử dụng bất biến vòng lặp trong để chứng minh cho vòng lặp ngoài
- Chứng minh bước **Dừng** cho bất biến vòng lặp ngoài

Thuật toán

Sắp xếp nổi bọt



Thuật toán 5: Sắp xếp nổi bọt (Bubble Sort)

Input: $a_1, a_2, \dots, a_n$: dãy số thực ($n \geq 2$)

Output: Dãy đã cho được sắp xếp theo thứ tự tăng dần

```
1 for i := 1 to n - 1 do
2   for j := 1 to n - i do
3     if  $a_j > a_{j+1}$  then
4       Hoán đổi giá trị của  $a_j$  và  $a_{j+1}$  —  $t_4$  —  $t_3$  —  $t_2$  —  $t_1$ 
```

$$T(n) = t_1 = \sum_{i=1}^{n-1} \left[t_2^i + (\text{t.g. tăng } i \text{ và kiểm tra } i \leq n - 1) \right]$$

$$t_2^i = \sum_{j=1}^{n-i} \left[t_3^j + (\text{t.g. tăng } j \text{ và kiểm tra } j \leq n - i) \right]$$

t_4 là $O(1)$ ($v := a_j, a_j := a_{j+1}, a_{j+1} := v$)

$T(n)$ là $O(n^2)$

(xấu nhất)

$t_3^j = t_4 + (\text{thời gian kiểm tra } a_j > a_{j+1})$

$T(n)$ là $O(n^2)$

(tốt nhất)

Thuật toán 1

Hoàng Anh Đức

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp

Tài liệu tham khảo

12

19



Thuật toán 10: Sắp xếp chèn (Insertion Sort)

Input: $a_1, a_2, \dots, a_n$: dãy số thực ($n \geq 2$)

Output: Dãy đã cho được sắp xếp theo thứ tự tăng dần

1 **for** $i = 2$ **to** n **do**

2 $m := a_i$ // m sắp được chèn vào dãy $a_1, \dots, a_{i-1}$

3 $j := i - 1$

4 **while** $j \geq 1$ **và** $m < a_j$ **do** // Nếu $m < a_j$, đẩy a_j
 sang phải để có chỗ chèn m

5 $a_{j+1} := a_j$

6 $j := j - 1$

7 $a_{j+1} := m$ // Chèn m

8 // Dãy $a_1, \dots, a_i$ đã được sắp thứ tự

Thuật toán

Sắp xếp chèn



Thuật toán I

Hoàng Anh Đức

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp

Tài liệu tham khảo

Thuật toán 10 có hai vòng lặp: vòng lặp trong **while** ở Dòng 4–6 và vòng lặp ngoài **for** (chứa vòng lặp trong) ở Dòng 1–7

- Một bất biến vòng lặp cho vòng lặp ngoài là

Ở trước lần lặp i , dãy $a_1, \dots, a_{i-1}$ là dãy tăng

- Một bất biến vòng lặp cho vòng lặp trong là

Ở trước lần lặp j , $m \leq \min\{a_{j+1}, \dots, a_i\}$

14

19

Thuật toán

Sắp xếp chèn



Thuật toán I

Hoàng Anh Đức

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp

Tài liệu tham khảo

Thuật toán 6: Sắp xếp chèn (Insertion Sort)

Input: $a_1, a_2, \dots, a_n$: dãy số thực ($n \geq 2$)

Output: Dãy đã cho được sắp xếp theo thứ tự tăng dần

```
1 for  $i = 2$  to  $n$  do
2    $m := a_i$ 
3    $j := i - 1$ 
4   while  $j \geq 1$  và  $m < a_j$  do
5      $a_{j+1} := a_j$ 
6      $j := j - 1$ 
7    $a_{j+1} := m$ 
```

Diagram illustrating the execution of Insertion Sort with time steps t_2, t_3, t_4, t_5, t_6 and a red bracket t_1 indicating the main loop.

$$T(n) = t_1 = \sum_{i=2}^n \left[\sum_{k=2}^5 t_k^i + (\text{t.g. tăng } i \text{ và kiểm tra } i \leq n) \right]$$

$t_2^i, t_3^i, t_5^i, t_6^{i,j}$ là $O(1)$

$$t_4^i = \sum_{1 \leq j \leq i-1 \text{ và } m < a_j} [t_6^{i,j} + (\text{t.g. kiểm tra } j \geq 1 \text{ và } m < a_j)]$$

$T(n)$ là $O(n^2)$

(xấu nhất)

$T(n)$ là $O(n)$

(tốt nhất)

15

19

Một số lỗi thường gặp



Thuật toán I

Hoàng Anh Đức

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

16 Một số lỗi thường gặp

Tài liệu tham khảo

Chú ý

Tham khảo từ tài liệu “Common Mistakes in Discrete Mathematics” (https://highered.mheducation.com/sites/dl/free/125967651x/1106131/Common_Mistakes_in_Discrete_Math.pdf)

- (a) **Quên tăng biến đếm bên trong vòng lặp của các thủ tục khi xây dựng thuật toán**
 - Nếu biến đếm không được tăng, vòng lặp có thể sẽ chạy mãi mãi
- (b) **Nhầm lẫn giữa ước lượng O -lớn và Θ -lớn, tức là nghĩ rằng O -lớn cung cấp cả chặn dưới và chặn trên cho kích thước của hàm**
 - Ước lượng O -lớn chỉ là chặn trên

Một số lỗi thường gặp (tiếp)



Thuật toán I

Hoàng Anh Đức

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

17

Một số lỗi thường gặp

Tài liệu tham khảo

- Không chính xác khi nói rằng một thuật toán kém hiệu quả vì nó chạy trong thời gian $O(n^7)$, bởi vì, ví dụ, một thuật toán chạy trong thời gian tuyến tính (có thời gian chạy tỉ lệ với n) cũng thỏa mãn ước lượng này
- Trong trường hợp này, điều chính xác cần nói (nếu đúng) là thuật toán chạy trong thời gian $\Theta(n^7)$

(c) Không hiểu rằng hằng số trong ước lượng O -lớn có thể rất lớn

- Ví dụ, nếu một thuật toán có thời gian chạy $O(n^2)$, thì nó có thể mất $10^{55}n^2$ bước và do đó không khả thi ngay cả với các giá trị n nhỏ

(d) Bị nhầm lẫn bởi số mũ khi so sánh các hàm khác nhau

- Ví dụ, thoạt nhìn có vẻ như $(\log n)^{100}$ tăng nhanh hơn $n^{1/2}$, nhưng thực tế $n^{1/2}$ tăng nhanh hơn, vì $\log n$ tăng rất chậm so với n

Một số lỗi thường gặp (tiếp)



Thuật toán I
Hoàng Anh Đức

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

18 Một số lỗi thường gặp

Tài liệu tham khảo

- Vì thế $(\log n)^{100}$ là $O(n^{1/2})$, nhưng $n^{1/2}$ không phải là $O((\log n)^{100})$

(e) Cộng thay vì nhân, hoặc ngược lại, khi phân tích thời gian chạy của thuật toán

- Nếu một vòng lặp lồng bên trong một vòng lặp khác, thì thời gian chạy được nhân lên
- Nếu một vòng lặp đi sau vòng lặp khác, thời gian chạy được cộng lại

Tài liệu tham khảo



Thuật toán I

Hoàng Anh Đức

Tìm kiếm và Sắp xếp

Một số thuật toán tìm kiếm

Tìm kiếm nhị phân

Một số thuật toán sắp xếp

Sắp xếp nổi bọt

Sắp xếp chèn

Một số lỗi thường gặp



Rosen, Kenneth (2012). *Discrete Mathematics and Its Applications*. 7th. McGraw-Hill.

19

Tài liệu tham khảo

1. Logic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Đệ quy
4. Thuật toán I: Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp
5. Thuật toán II: Thuật toán tham lam, thuật toán đệ quy
6. Lý thuyết số cơ bản
7. Các phương pháp đếm
8. Lý thuyết đồ thị I: Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông
9. Lý thuyết đồ thị II: Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị
10. Lý thuyết đồ thị III: Cây
11. Đại số Boole

VNU-HUS MAT3500: Toán rời rạc

Thuật toán II

Thuật toán đệ quy, thuật toán tham lam

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn



Nội dung



Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

Thuật toán đệ quy

Giới thiệu



Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

2

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thơ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

- Định nghĩa theo đệ quy không những có thể áp dụng cho các hàm và tập hợp mà còn cho cả các thuật toán
- Một *thủ tục đệ quy (recursive procedure)* là một thủ tục gọi chính nó
- Một *thuật toán đệ quy (recursive algorithm)* là một thuật toán giải một bài toán bằng cách chuyển về việc giải chính bài toán đó nhưng với đầu vào có kích thước nhỏ hơn
 - *Kỹ thuật chia để trị (divide-and-conquer technique)*: giải một bài toán ban đầu thông qua việc chia nó thành các bài toán nhỏ hơn cùng loại và giải chúng

Thuật toán đệ quy

Giới thiệu



Thuật toán II

Hoàng Anh Đức

Một thuật toán đệ quy thường có cấu trúc như sau:

Trường hợp cơ sở Giải trực tiếp bài toán với đầu vào có kích thước đủ nhỏ hoặc đầu vào đặc biệt mà không cần gọi đệ quy

Chia thành các bài toán con Với đầu vào không thuộc trường hợp cơ sở, thuật toán chia bài toán thành một hoặc nhiều **bài toán con** có kích thước nhỏ hơn

Giải các bài toán con Thuật toán gọi đệ quy chính nó để giải các bài toán con đã được định nghĩa

Kết hợp các lời giải Thuật toán kết hợp các lời giải của các bài toán con để tạo ra lời giải cho bài toán ban đầu

Thuật toán đệ quy

3

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

Thuật toán đệ quy

Giới thiệu



Thuật toán II
Hoàng Anh Đức

Chứng minh tính đúng đắn của thuật toán đệ quy bằng quy nạp mạnh

- **Phát biểu điều cần chứng minh:** Một điểm quan trọng là cần *chỉ rõ "thuật toán đúng" nghĩa là gì*
- **Bước cơ sở:** Các trường hợp khi *thuật toán cho ra kết quả một cách trực tiếp* mà không cần thông qua gọi đệ quy chính nó là các trường hợp cần xét trong bước cơ sở
 - Sử dụng mô tả của thuật toán để chỉ ra thuật toán sẽ trả lại gì trong trường hợp cơ sở
 - Chỉ ra rằng giá trị trả lại của thuật toán là đúng
- **Bước quy nạp:** Giả thiết rằng thuật toán đúng cho *mọi đầu vào kích thước nhỏ hơn*. Chỉ ra rằng thuật toán cũng đúng cho đầu vào hiện tại
 - Phát biểu giả thiết quy nạp: Giả sử thuật toán đúng với mọi đầu vào giữa trường hợp cơ sở và các đầu vào có kích thước nhỏ hơn một đơn vị so với đầu vào hiện tại
 - Mô tả cụ thể thuật toán trả lại gì với đầu vào hiện tại dựa trên các lần gọi đệ quy
 - Sử dụng giả thiết quy nạp để thay mỗi lần gọi đệ quy bằng đáp án chính xác. Chỉ ra rằng những điều này dẫn tới đáp án đúng cho trường hợp hiện tại
 - Nếu bạn xét nhiều trường hợp trong thuật toán thì cần thực hiện hai điều trên với từng trường hợp một

Thuật toán đệ quy

4

Giới thiệu

Tính giai thừa và lũy thừa
Tìm kiếm tuyến tính
Tìm kiếm nhị phân
Sắp xếp nổi bọt
Sắp xếp chèn
Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu
Định lý thợ
Cây đệ quy

Thuật toán tham lam

Giới thiệu
Bài toán Lập lịch

Thuật toán đệ quy

Giới thiệu



Thuật toán II
Hoàng Anh Đức

Phân tích thời gian chạy của thuật toán đệ quy:

■ *Thiết lập hệ thức truy hồi*

- Biểu diễn thời gian chạy $T(n)$ dưới dạng các hàm của thời gian chạy cho các bài toán con

■ *Giải hoặc ước lượng hệ thức truy hồi*

- Giải hệ thức truy hồi (đã đề cập trong phần “Quy nạp và Đệ quy”)
 - Đoán nghiệm và chứng minh bằng quy nạp
 - Sử dụng hàm đặc trưng
 - Sử dụng hàm sinh
- Ước lượng hệ thức truy hồi (sẽ đề cập trong phần sau)
 - Định lý thợ (Master Theorem)
 - Cây đệ quy (Recursion Tree)

5

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa
Tìm kiếm tuyến tính
Tìm kiếm nhị phân
Sắp xếp nổi bọt
Sắp xếp chèn
Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu
Định lý thợ
Cây đệ quy

Thuật toán tham lam

Giới thiệu
Bài toán Lập lịch

Thuật toán đệ quy

Tính giai thừa



Thuật toán II

Hoàng Anh Đức

Với mọi số nguyên không âm n

$$0! = 1$$

$$n! = n \cdot (n - 1)! \quad \forall n \geq 1$$

Thuật toán 1: Tính $n!$

Input: n : số nguyên không âm

Output: $n!$

```
1 procedure factorial( $n$ ):  
2   if  $n = 0$  then  
3     return 1  
4   else  
5     return  $n \cdot \text{factorial}(n - 1)$ 
```

Thuật toán đệ quy

Giới thiệu

6

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức
truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

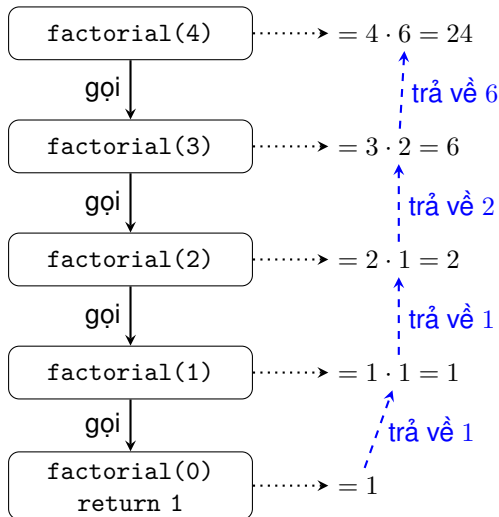
Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch



Ví dụ 1 (factorial(4))



Thuật toán đệ quy

Giới thiệu
Tính giai thừa và lũy thừa
Tìm kiếm tuyến tính
Tìm kiếm nhị phân
Sắp xếp nổi bọt
Sắp xếp chèn
Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu
Định lý thợ
Cây đệ quy

Thuật toán tham lam

Giới thiệu
Bài toán Lập lịch

Thuật toán đệ quy

Tính giai thừa



Thuật toán II

Hoàng Anh Đức

Ví dụ 2 (Thuật toán đệ quy tính giai thừa là đúng)

Ta chứng minh tính đúng đắn của Thuật toán 1 bằng quy nạp. Gọi $\text{factorial}(n)$ là giá trị trả lại bởi Thuật toán 1

- Ta chứng minh $\text{factorial}(n) = n!$ với mọi $n \geq 0$
- **Bước cơ sở:** Khi $n = 0$, $\text{factorial}(n) = 1 = n!$
- **Bước quy nạp:** Giả sử $\text{factorial}(k) = k!$ với số nguyên $k \geq 0$ nào đó. Ta chứng minh $\text{factorial}(k+1) = (k+1)!$.
Thật vậy, Thuật toán 1 trả lại
 $\text{factorial}(k+1) = (k+1) \cdot \text{factorial}(k)$. Theo giả thiết quy nạp, $\text{factorial}(k) = k!$. Do đó,
 $\text{factorial}(k+1) = (k+1) \cdot k! = (k+1)!$

Ví dụ 3 (Thời gian chạy của thuật toán đệ quy tính giai thừa)

$$T(n) = \max\{O(1), T(n-1) + O(1)\} + O(1) = T(n-1) + O(1)$$

nghĩa là tồn tại hằng số C thỏa mãn $T(n) = T(n-1) + C$. Suy ra
 $T(n) = O(n)$

Thuật toán đệ quy

Giới thiệu

8

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

Thuật toán đệ quy

Tính lũy thừa



Thuật toán II

Hoàng Anh Đức

Với mọi số thực $a \neq 0$ và số nguyên không âm n ,

$$a^0 = 1$$

$$a^n = a \cdot a^{n-1} \quad \forall n \geq 1$$

Thuật toán 2: Tính a^n

Input: a : số thực khác 0, n : số nguyên không âm

Output: a^n

procedure power(a, n):

if $n = 0$ **then**

return 1

else

return $a \cdot \text{power}(a, n - 1)$

Bài tập 1

Với Thuật toán 2, hãy

- Chứng minh tính đúng đắn
- Xây dựng hệ thức truy hồi để đánh giá thời gian chạy

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức
truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

Thuật toán đệ quy

Tính lũy thừa



Thuật toán II

Hoàng Anh Đức

Chú ý

Điều khó khăn nhất cần vượt qua khi suy nghĩ về các thuật toán đệ quy là sự miễn cưỡng không tin rằng trường hợp nhỏ hơn sẽ được xử lý đúng.

- Ví dụ, giả sử bạn muốn viết một thuật toán đệ quy để tính a^n , sử dụng các công thức $a^{2k} = (a^k)^2$ và $a^{2k+1} = (a^k)^2 \cdot a$.
- Lệnh gọi đệ quy sẽ xử lý việc tính toán a^k , vì vậy bạn không cần phải lo lắng về cách máy tính sẽ đệ quy nhiều lần, tất cả đường xuống đến trường hợp cơ sở, để thực hiện điều đó. *Miến là bước đệ quy và trường hợp cơ sở của bạn (ở đây là $a^0 = 1$) là đúng, thuật toán của bạn là đúng*

Thuật toán đệ quy

Giới thiệu
10 Tính giai thừa và lũy thừa
Tìm kiếm tuyến tính
Tìm kiếm nhị phân
Sắp xếp nổi bọt
Sắp xếp chèn
Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu
Định lý thợ
Cây đệ quy

Thuật toán tham lam

Giới thiệu
Bài toán Lập lịch

Thuật toán đệ quy

Tìm kiếm tuyến tính



Thuật toán II

Hoàng Anh Đức

Thuật toán 3: Tìm kiếm tuyến tính (Linear Search)

Input: $a_1, a_2, \dots, a_n$: dãy số nguyên, i, j, x : số nguyên,

$$1 \leq i \leq j \leq n$$

Output: Nếu $x \in \{a_i, a_{i+1}, \dots, a_j\}$ thì trả lại
 $k \in \{i, i+1, \dots, j\}$ sao cho $x = a_k$. Ngược lại thì
trả lại 0

```
1 procedure LinearSearch( $i, j, x$ ):
2   if  $a_i = x$  then // Ở đúng vị trí? Trả lại kết quả
3     return  $i$ 
4   else
5     if  $i = j$  then // Không tìm thấy
6       return 0
7     else // Tìm trong phần còn lại
8       return LinearSearch( $i + 1, j, x$ )
```

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

11

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức
truy hồi

Giới thiệu

Định lý thặng dư

Cây đệ quy

Thuật toán tham lam

Giới thiệu

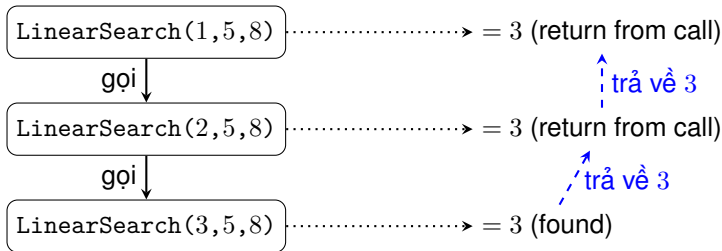
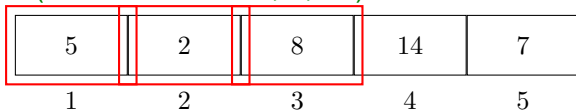
Bài toán Lập lịch

Thuật toán đệ quy

Tìm kiếm tuyến tính



Ví dụ 4 (LinearSearch(1, 5, 8))



Bài tập 2

Với Thuật toán 3, hãy

- Chứng minh tính đúng đắn
- Xây dựng hệ thức truy hồi để đánh giá thời gian chạy

Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

12 Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

Thuật toán đệ quy

Tìm kiếm nhị phân



Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

13 Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

Thuật toán 4: Tìm kiếm nhị phân (Binary Search)

Input: $a_1, a_2, \dots, a_n$: dãy số nguyên thực sự tăng, i, j, x : số nguyên, $1 \leq i \leq j \leq n$

Output: Nếu $x \in \{a_i, a_{i+1}, \dots, a_j\}$ thì trả lại $k \in \{i, i+1, \dots, j\}$ sao cho $x = a_k$. Ngược lại thì trả lại 0

```
1 procedure BinarySearch( $i, j, x$ ):
2    $m := \lfloor (i + j) / 2 \rfloor$  // Đi đến giữa dãy
3   if  $x = a_m$  then // Đúng vị trí?
4     return  $m$ 
5   else
6     if  $x < a_m$  và  $i < m$  then //  $x$  ở nửa bên trái?
7       return BinarySearch( $i, m - 1, x$ )
8     else
9       if  $x > a_m$  và  $j > m$  then //  $x$  ở nửa bên phải?
10        return BinarySearch( $m + 1, j, x$ )
11      else // Không tìm thấy
12        return 0
```

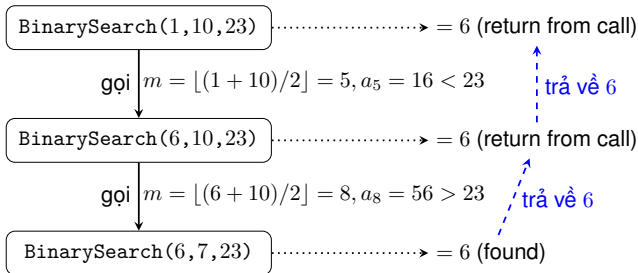
Thuật toán đệ quy

Tìm kiếm nhị phân



Ví dụ 5 (BinarySearch(1, 10, 23))

2	5	8	12	16	23	38	56	72	91
1	2	3	4	5	6	7	8	9	10



$$m = \lfloor (6 + 7)/2 \rfloor = 6, a_6 = 23 = 23$$

Bài tập 3

Xây dựng hệ thức truy hồi để đánh giá thời gian chạy của Thuật toán 4

Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

14

37

Thuật toán đệ quy

Sắp xếp nổi bọt



Thuật toán II

Hoàng Anh Đức

Thuật toán 5: Sắp xếp nổi bọt (Bubble Sort)

Input: $A = a_1, a_2, \dots, a_n$: dãy số nguyên, n : số nguyên dương

Output: Dãy số nguyên sắp thứ tự tăng dần

procedure BubbleSort(A, n):

```
1   if  $n = 1$  then
2       return  $A$  // Trường hợp cơ sở: mảng có 1
3       phần tử đã được sắp xếp
4   for  $i \leftarrow 1$  to  $n - 1$  do
5       if  $a_i > a_{i+1}$  then
6           Hoán đổi giá trị của  $a_i$  và  $a_{i+1}$ 
7   BubbleSort( $A, n - 1$ ) // Đệ quy với phần tử cuối
   cùng đã được đẩy đến đúng vị trí
```

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

15

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức
truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

Thuật toán đệ quy

Sắp xếp nổi bọt

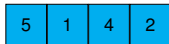


Thuật toán II

Hoàng Anh Đức

Ví dụ 6 (BubbleSort([5, 1, 4, 2], 4))

Mảng ban đầu:



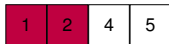
BubbleSort([5, 1, 4, 2], 4)

Lần lượt so sánh: [5, 1], [1, 4], [4, 2]



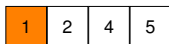
BubbleSort([1, 4, 2, 5], 3)

Lần lượt so sánh: [1, 4], [4, 2]



BubbleSort([1, 2, 4, 5], 2)

Lần lượt so sánh: [1, 2]



BubbleSort([1, 2, 4, 5], 1)

Trường hợp cơ sở: $n = 1$, mảng đã sắp xếp

Bài tập 4

Xây dựng hệ thức truy hồi để đánh giá thời gian chạy của Thuật toán 5

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

16

37



Thuật toán 6: Sắp xếp chèn (Insertion Sort)

Input: $A = a_1, a_2, \dots, a_n$: dãy số nguyên, n : số nguyên dương

Output: Dãy số nguyên sắp thứ tự tăng dần

```
1 procedure InsertionSort( $A, n$ ):
2   if  $n \leq 1$  then
3     return  $A$       // Trường hợp cơ sở: mảng có 0
                     // hoặc 1 phần tử đã được sắp xếp
4   InsertionSort( $A, n - 1$ )  // Sắp xếp đệ quy các
                             // phần tử từ  $a_1$  đến  $a_{n-1}$ 
5   // Chèn phần tử  $a_n$  vào vị trí thích hợp trong
   // mảng đã sắp xếp
6    $key := a_n$ 
7    $i := n - 1$ 
8   while  $i \geq 1$  và  $a_i > key$  do
9      $a_{i+1} := a_i$ 
10     $i := i - 1$ 
```

Thuật toán đệ quy

Sắp xếp chèn



Ví dụ 7 (InsertionSort([5, 2, 4, 1], 4))

Mảng ban đầu:

5	2	4	1
---	---	---	---

InsertionSort([5, 2, 4, 1], 4)

gọi ↓

InsertionSort([5, 2, 4, 1], 3)

gọi ↓

InsertionSort([5, 2, 4, 1], 2)

gọi ↓

InsertionSort([5, 2, 4, 1], 1)

Sắp xếp đệ quy [5, 2, 4], sau đó chèn 1

Sắp xếp đệ quy [5, 2], sau đó chèn 4

Sắp xếp đệ quy [5], sau đó chèn 2

Trường hợp cơ sở: mảng [5] đã sắp xếp

2	5	4	1
---	---	---	---

Chèn 2 vào [5]

2	4	5	1
---	---	---	---

Chèn 4 vào [2, 5]

1	2	4	5
---	---	---	---

Chèn 1 vào [2, 4, 5]

Kết quả cuối cùng

Bài tập 5

Xây dựng hệ thức truy hồi để đánh giá thời gian chạy của Thuật toán 6

Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

18

37

Thuật toán đệ quy

Sắp xếp trộn



Thuật toán II

Hoàng Anh Đức

Thuật toán 7: Sắp xếp trộn (Merge Sort)

Input: $L = a_1, a_2, \dots, a_n$: dãy số nguyên

Output: Dãy số nguyên sắp thứ tự tăng dần

```
1 procedure MergeSort( $L$ ):  
2   if  $n > 1$  then  
3      $m := \lfloor n/2 \rfloor$   
4      $L_1 := a_1, \dots, a_m; L_2 := a_{m+1}, \dots, a_n$   
5      $L := \text{Merge}(\text{MergeSort}(L_1), \text{MergeSort}(L_2))$   
  
6 procedure Merge( $A, B$ ):  
   Input:  $A = (a_1, \dots, a_{|A|}), B = (b_1, \dots, b_{|B|})$ : dãy số đã sắp thứ tự  
   Output: Dãy các số trong cả  $A$  và  $B$  sắp thứ tự tăng dần  
   if  $A = \emptyset$  then  
7     return  $B$   
   if  $B = \emptyset$  then  
9     return  $A$   
   if  $a_1 < b_1$  then  
11    return  $(a_1, \text{Merge}(a_2, \dots, a_{|A|}, B))$   
   else  
12    return  $(b_1, \text{Merge}(A, b_2, \dots, b_{|B|}))$   
13
```

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

19

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

37

Thuật toán đệ quy

Sắp xếp trộn



Thuật toán II

Hoàng Anh Đức

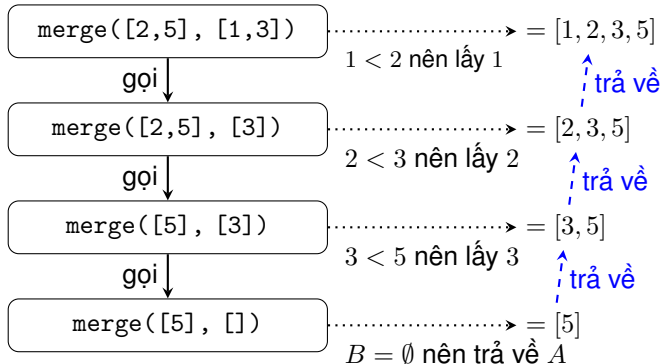
Ví dụ 8 (Merge([2, 5], [1, 3]))

Mảng A:

2	5
---	---

Mảng B:

1	3
---	---



Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

20

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

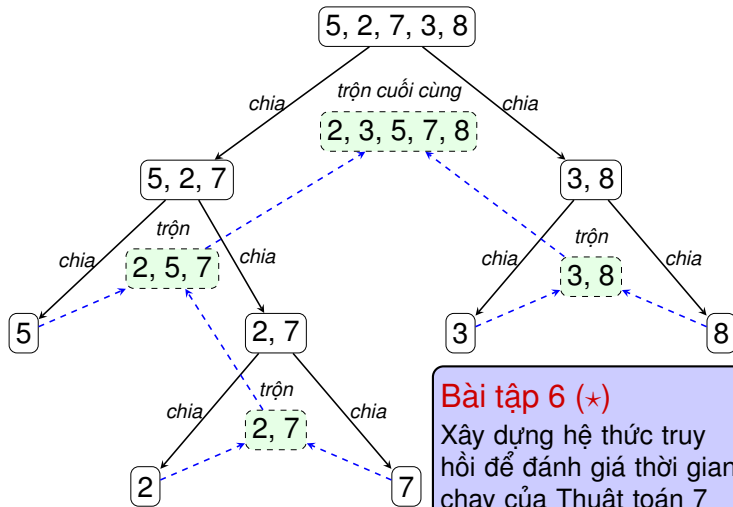
37

Thuật toán đệ quy

Sắp xếp trộn



Ví dụ 9 (MergeSort([5, 2, 7, 3, 8]))



Bài tập 6 (★)

Xây dựng hệ thức truy hồi để đánh giá thời gian chạy của Thuật toán 7

Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

21

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

37

Ước lượng hệ thức truy hồi

Giới thiệu



Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức
truy hồi

22

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

- Hệ thức truy hồi thường xuất hiện khi phân tích thuật toán đệ quy
- Ước lượng hệ thức truy hồi giúp ta xác định độ phức tạp thời gian của thuật toán
- Hai phương pháp chính để ước lượng hệ thức truy hồi:
 - (1) **Định lý thợ (Master Theorem)**: Công cụ mạnh mẽ giúp ước lượng nhanh chóng các hệ thức có dạng
$$f(n) = af(n/b) + cn^d$$
 - (2) **Cây đệ quy (Recursion Tree)**: Biểu diễn trực quan quá trình tính toán

Ước lượng hệ thức truy hồi

Định lý thợ



Thuật toán II

Hoàng Anh Đức

Định lý 1: Định lý thợ (Master Theorem)

Gọi f là một hàm tăng thỏa mãn hệ thức truy hồi

$$f(n) = af(n/b) + cn^d$$

trong đó $n = b^k$ với k là số nguyên dương nào đó, $a \geq 1$, b là số nguyên dương lớn hơn 1, và c, d là các số thực với c dương và d không âm. Ta có

$$f(n) \text{ là } \begin{cases} O(n^d) & \text{nếu } a < b^d \\ O(n^d \log n) & \text{nếu } a = b^d \\ O(n^{\log_b a}) & \text{nếu } a > b^d \end{cases}$$

Ví dụ 10

- Với $T(n) = 2T(n/2) + n$, ta có $T(n) = O(n \log n)$
- Với $T(n) = T(n/2) + n$, ta có $T(n) = O(n)$
- Với $T(n) = 3T(n/2) + n$, ta có $T(n) = O(n^{\log 3})$

Thuật toán đệ quy

Giới thiệu
Tính giai thừa và lũy thừa
Tìm kiếm tuyến tính
Tìm kiếm nhị phân
Sắp xếp nổi bọt
Sắp xếp chèn
Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu
Định lý thợ
Cây đệ quy

Thuật toán tham lam

Giới thiệu
Bài toán Lập lịch

23

37

Ước lượng hệ thức truy hồi

Định lý thợ



Thuật toán II

Hoàng Anh Đức

Bài tập 7

Sử dụng Định lý thợ (Định lý 1), hãy ước lượng các hệ thức truy hồi sau theo O -lớn, giả sử $T(1) = 1$

(a) $T(n) = 4T(n/3) + n^2$

(b) $T(n) = 4T(n/2) + n^2$

(c) $T(n) = 3T(n/3) + n \log n$

(d) $T(n) = 3T(n/3) + 1$

(e) $T(n) = 2T(n/2) + n^n$

(f) $T(n) = 7T(n/2) + n^2$

(g) $T(n) = 16T(n/4) + n^2$

(h) $T(n) = 2T(n/2) - n^2$

(i) $T(n) = 8T(n/2) + n^3$

(j) $T(n) = 3T(n/3) + \sqrt{n}$

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

24

37

Ước lượng hệ thức truy hồi

Cây đệ quy



Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

- **Cây đệ quy (Recursion Tree)** là biểu diễn trực quan của quá trình chạy thuật toán đệ quy
- Mỗi nút trong cây biểu diễn **chi phí (cost)** của một **lần gọi đệ quy (recursive call)**
- **Các bước xây dựng và phân tích cây đệ quy:**
 - (1) Vẽ cây với gốc ứng với lần gọi ban đầu $T(n)$
 - (2) Phân tách mỗi nút thành các chi phí: chi phí không đệ quy (non-recursive cost) và các lần gọi đệ quy (recursive calls)
 - (3) Tiếp tục phân tách cho đến khi đạt trường hợp cơ sở (base case)
 - (4) Tính tổng chi phí theo từng mức (level) của cây
 - (5) Tính tổng chi phí các mức để có kết quả cuối cùng
- **Ưu điểm:** Trực quan, giúp hiểu được bản chất của thuật toán và áp dụng được với nhiều dạng hệ thức mà Định lý thợ không áp dụng được
- **Nhược điểm:** Không phải lúc nào cũng dễ vẽ và phân tích được

25

37

Ước lượng hệ thức truy hồi

Cây đệ quy

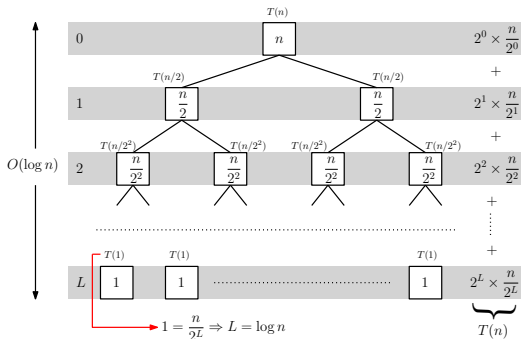


Thuật toán II

Hoàng Anh Đức

Ví dụ 11

Xét hệ thức truy hồi $T(n) = 2T(n/2) + n$ với điều kiện ban đầu $T(1) = 1$ và $n = 2^k$ với số nguyên $k \geq 1$ nào đó. Ta vẽ cây đệ quy cho hệ thức này như sau



$$\text{Ta có } T(n) = \sum_{i=0}^L 2^i \cdot \frac{n}{2^i} = n(\log n + 1) = O(n \log n)$$

Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ

26

Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

37

Ước lượng hệ thức truy hồi

Cây đệ quy

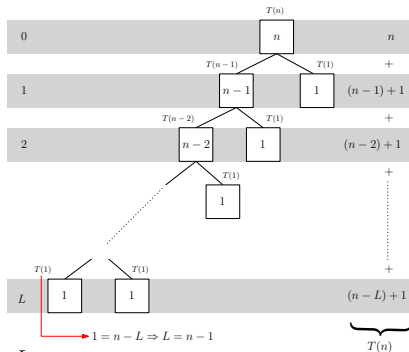


Thuật toán II

Hoàng Anh Đức

Ví dụ 12

Xét hệ thức truy hồi $T(n) = T(n-1) + T(1) + n$ với $n \geq 1$ và điều kiện ban đầu $T(1) = 1$. Ta vẽ cây đệ quy của hệ thức này như sau



$$\text{Ta có } T(n) \leq \sum_{i=0}^L n = O(n^2)$$

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

27

37

Ước lượng hệ thức truy hồi

Cây đệ quy



Thuật toán II

Hoàng Anh Đức

Bài tập 8

Sử dụng cây đệ quy để ước lượng $T(n)$ cho bởi các hệ thức truy hồi sau

(a) $T(n) = 2T(n/2) + n^2$

(b) $T(n) = T(n/2) + 1$

(c) $T(n) = 2T(n-1) + 1$

(d) (*) $T(n) = 2T(n/2) + n \log n$ (**Đáp án:** $O(n \log^2 n)$)

Bài tập 9

Chứng minh Định lý thợ bằng cách sử dụng cây đệ quy

(a) Vẽ cây đệ quy cho $T(n) = aT(n/b) + cn^d$ trong đó $n = b^k$ với k là số nguyên dương nào đó, $a \geq 1$, b là số nguyên dương lớn hơn 1, và c, d là các số thực với c dương và d không âm

(b) Tính tổng từng hàng và viết công thức của $T(n)$ dưới dạng tổng của các hàng trong cây.

(c) Xét các trường hợp $a < b^d$, $a = b^d$, và $a > b^d$

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

28

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

37

Ước lượng hệ thức truy hồi

Cây đệ quy



Thuật toán II

Hoàng Anh Đức

Bài tập 10

Xây dựng hệ thức truy hồi cho thời gian chạy của các thuật toán đệ quy đã đề cập trong bài giảng. Giải hoặc ước lượng hệ thức bạn tìm được để chứng minh thời gian chạy của các thuật toán như sau:

- Tính lũy thừa (Thuật toán 2): $T(n) = O(n)$
- Tìm kiếm tuyến tính (Thuật toán 3): $T(n) = O(n)$
- Tìm kiếm nhị phân (Thuật toán 4): $T(n) = O(\log n)$
- Sắp xếp nổi bọt (Thuật toán 5): $T(n) = O(n^2)$
- Sắp xếp chèn (Thuật toán 6): $T(n) = O(n^2)$
- Sắp xếp trộn (Thuật toán 7): $T(n) = O(n \log n)$

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

29

37

Thuật toán tham lam

Giới thiệu



Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

- **Các bài toán tối ưu (optimization problems)** yêu cầu cực đại hóa hoặc cực tiểu hóa một số tham số xét trên tập tất cả các đầu vào có thể
 - Tìm đường đi giữa hai thành phố với khoảng cách **nhỏ nhất**
 - Tìm cách đặt các trạm phát sóng sao cho diện tích phủ sóng **lớn nhất**
- Một **thuật toán tham lam (greedy algorithm)** thường được sử dụng để giải bài toán tối ưu: luôn chọn biện pháp “tốt nhất” ở mỗi bước địa phương (theo một số tiêu chuẩn cục bộ nào đó) với hi vọng sẽ thu được một lời giải tối ưu trên toàn cục
 - Giải thuật này không nhất thiết xuất ra một lời giải tối ưu cho toàn bộ bài toán, nhưng trong nhiều trường hợp cụ thể nó có thể xuất ra lời giải tối ưu
 - Sau khi mô tả cụ thể “lựa chọn tốt nhất ở từng bước”, ta cố gắng chứng minh rằng giải thuật này luôn cho ta một lời giải tối ưu hoặc tìm một phản ví dụ để chỉ ra điều ngược lại

30

37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

Giới thiệu
Tính giai thừa và lũy thừa
Tìm kiếm tuyến tính
Tìm kiếm nhị phân
Sắp xếp nổi bọt
Sắp xếp chèn
Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu
Định lý thợ
Cây đệ quy

Thuật toán tham lam

Giới thiệu
Bài toán Lập lịch

■ Bài toán:

■ Input:

- Một nhóm các bài giảng với thời gian bắt đầu và kết thúc
- Chỉ có một giảng đường duy nhất
- Khi một bài giảng bắt đầu, nó tiếp diễn cho đến khi kết thúc
- Không có hai bài giảng nào được tiến hành ở cùng thời điểm
- Ngay sau khi một bài giảng kết thúc, một bài giảng khác có thể bắt đầu

■ Output: Một danh sách các bài giảng dài nhất có thể

- Ở đây, nếu ta muốn áp dụng giải thuật tham lam, làm thế nào để “lựa chọn tốt nhất” ở mỗi bước của thuật toán? Nói cách khác, ta sẽ chọn bài giảng như thế nào?

- (1) Chọn bài giảng bắt đầu sớm nhất trong số các bài giảng bắt đầu sau các bài giảng ta vừa chọn trước đó?
- (2) Chọn bài giảng ngắn nhất trong số các bài giảng bắt đầu sau các bài giảng ta vừa chọn trước đó?
- (3) Chọn bài giảng kết thúc sớm nhất trong số các bài giảng bắt đầu sau các bài giảng ta vừa chọn trước đó?

31

37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

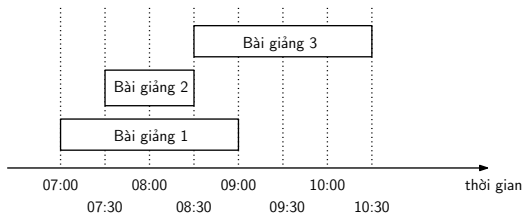
Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

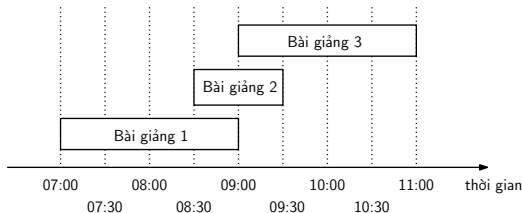
Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

(1) Chọn bài giảng bắt đầu sớm nhất? ❌



(2) Chọn bài giảng ngắn nhất? ❌



(3) Chọn bài giảng kết thúc sớm nhất? ✅

32

37



Thuật toán 8: Lập lịch tham lam (Greedy Scheduling)

Input: $(s_1, e_1), (s_2, e_2), \dots, (s_n, e_n)$: thời gian bắt đầu và kết thúc bài giảng $b_1, b_2, \dots, b_n$

Output: Danh sách bài giảng S có số bài giảng lớn nhất trong đó không có hai bài giảng nào xung đột nhau

- 1 Sắp xếp các bài giảng theo thứ tự tăng dần theo thời gian kết thúc và gán lại nhãn bài giảng sao cho

$$e_1 \leq e_2 \leq \dots \leq e_n$$

- 2 $S := \emptyset$

- 3 **for** $j := 1$ **to** n **do**

- 4 **if** Bài giảng j không xung đột với các phần tử của S
 then

- 5 $S := S \cup \{j\}$

- 6 **return** S

Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

Thuật toán tham lam

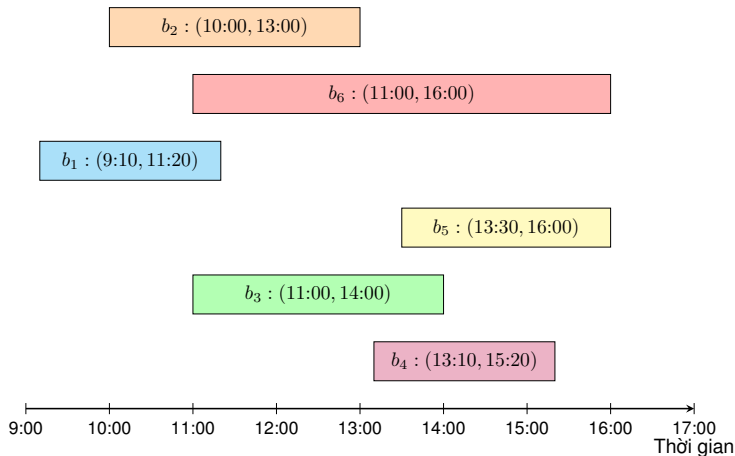
Bài toán Lập lịch



Thuật toán II
Hoàng Anh Đức

Ví dụ 13

Các bài giảng ban đầu:



Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

34

37

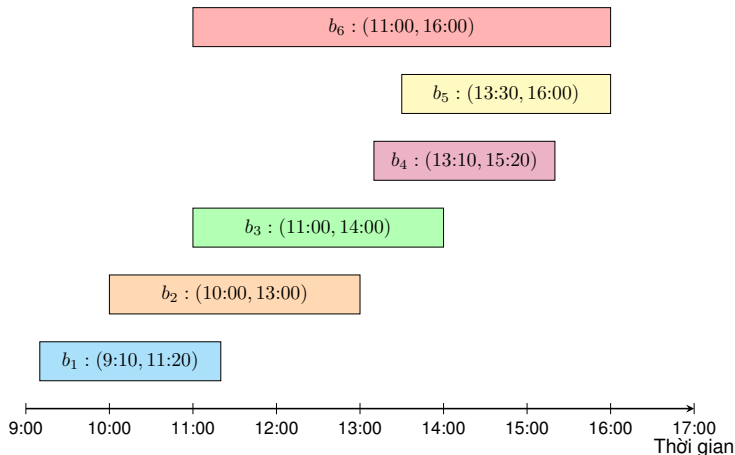
Thuật toán tham lam

Bài toán Lập lịch



Ví dụ 13

Bước 1: Sắp xếp theo thời gian kết thúc



Thuật toán II

Hoàng Anh Đức

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thợ

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

34

37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II

Hoàng Anh Đức

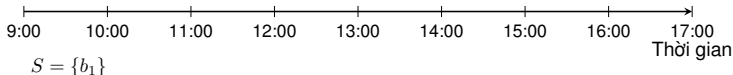
Ví dụ 13

Bước 2: Khởi tạo $S = \emptyset$

Bước 3: Duyệt các bài giảng

Xét $j = 1$: Thêm b_1 vào S

$b_1 : (9:10, 11:20)$



Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

34

37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II

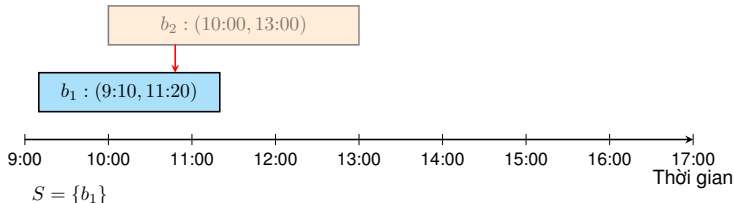
Hoàng Anh Đức

Ví dụ 13

Bước 2: Khởi tạo $S = \emptyset$

Bước 3: Duyệt các bài giảng

Xét $j = 2$: Không thêm b_2 vì xung đột với b_1



Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

34

37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II

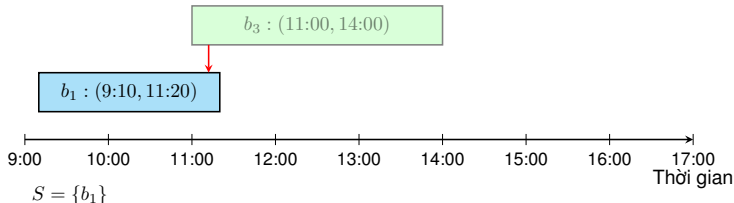
Hoàng Anh Đức

Ví dụ 13

Bước 2: Khởi tạo $S = \emptyset$

Bước 3: Duyệt các bài giảng

Xét $j = 3$: Không thêm b_3 vì xung đột với b_1



Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

34

37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II

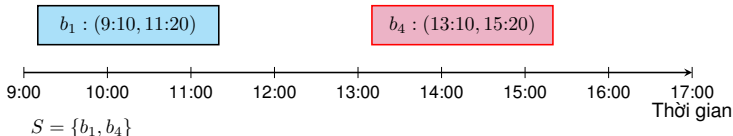
Hoàng Anh Đức

Ví dụ 13

Bước 2: Khởi tạo $S = \emptyset$

Bước 3: Duyệt các bài giảng

Xét $j = 4$: Thêm b_4 vào S



Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

34

37

Thuật toán tham lam

Bài toán Lập lịch



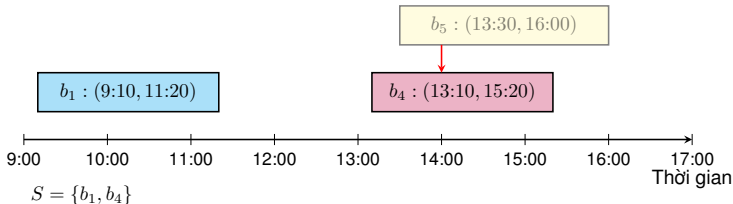
Thuật toán II
Hoàng Anh Đức

Ví dụ 13

Bước 2: Khởi tạo $S = \emptyset$

Bước 3: Duyệt các bài giảng

Xét $j = 5$: Không thêm b_5 vì xung đột với b_4



Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

34

37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II

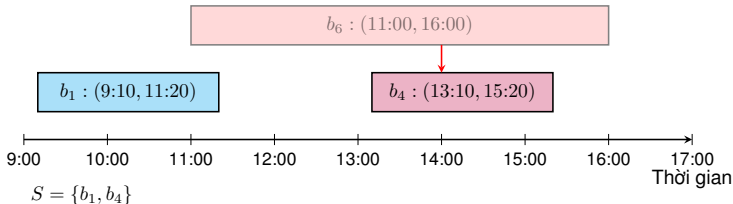
Hoàng Anh Đức

Ví dụ 13

Bước 2: Khởi tạo $S = \emptyset$

Bước 3: Duyệt các bài giảng

Xét $j = 6$: Không thêm b_6 vì xung đột với b_4



Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

34

37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II
Hoàng Anh Đức

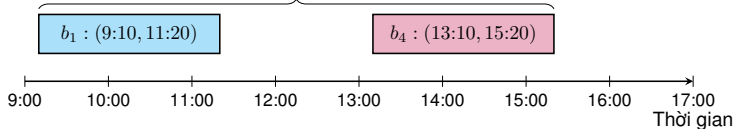
Ví dụ 13

Bước 2: Khởi tạo $S = \emptyset$

Bước 3: Duyệt các bài giảng

Kết quả: $S = \{b_1, b_4\}$

Lịch tối ưu với 2 bài giảng



Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

34

37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II

Hoàng Anh Đức

Định lý 2

Thuật toán 8 xuất ra một danh sách các bài giảng tối ưu

Chứng minh.

- Giả sử $S^* = (x_1, x_2, \dots, x_k)$ là danh sách tối ưu trong đó các bài giảng được sắp xếp theo thứ tự tăng dần theo thời gian kết thúc
- Giả sử $S = (y_1, y_2, \dots, y_{k'})$ là một danh sách bài giảng xuất ra từ Thuật toán 8 trong đó các bài giảng được sắp xếp theo thứ tự tăng dần theo thời gian kết thúc
- Do S^* là tối ưu, $k \geq k'$
- Nếu $S = S^*$ thì ta có điều phải chứng minh. Ngược lại, nếu $S \neq S^*$, gọi i là chỉ số đầu tiên trong $\{1, \dots, k'\}$ thỏa mãn $x_i \neq y_i$, nghĩa là

$$S^* = \langle x_1 = y_1, x_2 = y_2, \dots, x_{i-1} = y_{i-1}, \mathbf{x_i}, \dots, x_{k'}, \dots, x_k \rangle$$

$$S = \langle x_1 = y_1, x_2 = y_2, \dots, x_{i-1} = y_{i-1}, \mathbf{y_i}, \dots, y_{k'} \rangle$$

Thuật toán đệ quy

Giới thiệu

Tính giai thừa và lũy thừa

Tìm kiếm tuyến tính

Tìm kiếm nhị phân

Sắp xếp nổi bọt

Sắp xếp chèn

Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu

Định lý thặng dư

Cây đệ quy

Thuật toán tham lam

Giới thiệu

Bài toán Lập lịch

35

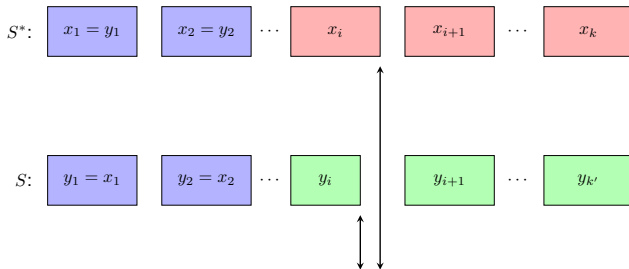
37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II
Hoàng Anh Đức



Nếu không tồn tại $i \in \{1, \dots, k'\}$ thỏa mãn $x_i \neq y_i$, ta có
 $S^* = \langle x_1 = y_1, x_2 = y_2, \dots, x_{k'} = y_{k'}, x_{k'+1}, \dots, x_k \rangle$
Ngược lại, thuật toán tham lam chọn y_i vì nó kết thúc sớm hơn x_i

Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

36

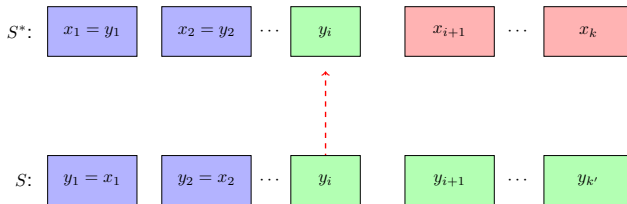
37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II
Hoàng Anh Đức



Thay thế x_i bằng y_i trong S^* vẫn giữ tính tối ưu
vì y_i kết thúc sớm hơn và không xung đột với $x_{i+1}, \dots, x_k$

Thuật toán đệ quy

- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch

36

37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II
Hoàng Anh Đức

Thuật toán đệ quy

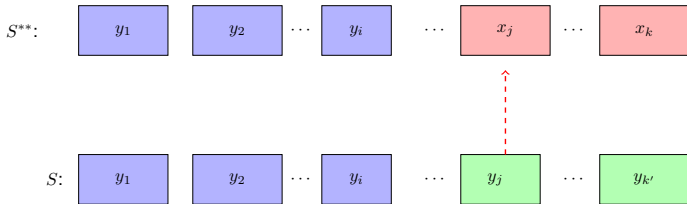
- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch



Thời gian

Sau khi thay thế x_i bằng y_i , ta thu được lời giải tối ưu mới S^{**} .
Tiếp đó, tìm j là chỉ số nhỏ nhất thỏa mãn $x_j \neq y_j$.
Lý luận tương tự để thay thế x_j bằng y_j .

36

37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II
Hoàng Anh Đức

Thuật toán đệ quy

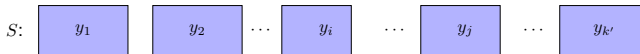
- Giới thiệu
- Tính giai thừa và lũy thừa
- Tìm kiếm tuyến tính
- Tìm kiếm nhị phân
- Sắp xếp nổi bọt
- Sắp xếp chèn
- Sắp xếp trộn

Ước lượng hệ thức truy hồi

- Giới thiệu
- Định lý thợ
- Cây đệ quy

Thuật toán tham lam

- Giới thiệu
- Bài toán Lập lịch



Thời gian

Tiếp tục quá trình này cho đến khi $S^* = \langle y_1, y_2, \dots, y_{k'}, x_{k'+1}, \dots, x_k \rangle$
Do thuật toán kết thúc khi chọn $y_{k'}$,
các phần tử $x_{k'+1}, \dots, x_k$ phải xung đột với $y_{k'}$ nên không thể tồn tại
Do đó $S^* = S = \langle y_1, y_2, \dots, y_{k'} \rangle$ là tối ưu

36

37

Thuật toán tham lam

Bài toán Lập lịch



Thuật toán II

Hoàng Anh Đức

Định lý 3

Thuật toán 8 chạy trong thời gian $O(n \log n)$, với n là số lượng bài giảng

Chứng minh.

Phân tích độ phức tạp:

- Sắp xếp n bài giảng theo thời gian kết thúc: $O(n \log n)$
 - Ví dụ, sử dụng Thuật toán 7 (Sắp xếp trộn)
- Vòng lặp **for** duyệt qua n bài giảng: $O(n)$
- Kiểm tra xung đột giữa bài giảng j với các bài giảng trong S :
 - Phương pháp đơn giản: $O(|S|) = O(n)$ trong trường hợp xấu nhất
 - Phương pháp tối ưu: Chỉ cần kiểm tra thời gian bắt đầu của bài giảng j với thời gian kết thúc của bài giảng cuối cùng trong S : $O(1)$

Thuật toán đệ quy

Giới thiệu
Tính giai thừa và lũy thừa
Tìm kiếm tuyến tính
Tìm kiếm nhị phân
Sắp xếp nổi bọt
Sắp xếp chèn
Sắp xếp trộn

Ước lượng hệ thức truy hồi

Giới thiệu
Định lý thợ
Cây đệ quy

Thuật toán tham lam

Giới thiệu
Bài toán Lập lịch

37

37

Part I

Phụ lục

Nội dung



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)



Định lý: (Nhắc lại Định lý 1)

Gọi f là một hàm tăng thỏa mãn hệ thức truy hồi

$$f(n) = af(n/b) + cn^d$$

trong đó $n = b^k$ với k là số nguyên dương nào đó, $a \geq 1$, b là số nguyên dương lớn hơn 1, và c, d là các số thực với c dương và d không âm. Ta có

$$f(n) \text{ là } \begin{cases} O(n^d) & \text{nếu } a < b^d \\ O(n^d \log n) & \text{nếu } a = b^d \\ O(n^{\log_b a}) & \text{nếu } a > b^d \end{cases}$$

2 Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)



Ta có

$$\begin{aligned}f(n) &= af(n/b) + cn^d \\&= a(af(n/b^2) + c(n/b)^d) + cn^d \\&= a^2 f(n/b^2) + cn^d(1 + \frac{a}{b^d}) \\&= a^2 (af(n/b^3) + c(n/b^2)^d) + cn^d(1 + \frac{a}{b^d}) \\&= a^3 f(n/b^3) + cn^d(1 + \frac{a}{b^d} + \left(\frac{a}{b^d}\right)^2) \\&= \dots \\&= a^k f(n/b^k) + cn^d \sum_{i=0}^{k-1} \left(\frac{a}{b^d}\right)^i\end{aligned}$$

3 Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Xét phương trình $f(n) = a^k f(n/b^k) + cn^d \sum_{i=0}^{k-1} \left(\frac{a}{b^d}\right)^i$

(1) Nếu $a = b^d$, ta có $\sum_{i=0}^{k-1} \left(\frac{a}{b^d}\right)^i = k$

(2) Nếu $a < b^d$, ta có $\sum_{i=0}^{k-1} \left(\frac{a}{b^d}\right)^i \leq \sum_{i=0}^{\infty} \left(\frac{a}{b^d}\right)^i = \frac{1}{1 - \frac{a}{b^d}}$. Do

a, b, d là các hằng số, ta có $\sum_{i=0}^{k-1} \left(\frac{a}{b^d}\right)^i = O(1)$

(3) Nếu $a > b^d$, ta có $\sum_{i=0}^{k-1} \left(\frac{a}{b^d}\right)^i = \frac{(a/b^d)^k - 1}{a/b^d - 1} = O((a/b^d)^k)$

4 Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Do đó,

$$f(n) = a^k f(n/b^k) + cn^d \cdot \begin{cases} O(k) & \text{nếu } a = b^d \\ O(1) & \text{nếu } a < b^d \\ O((a/b^d)^k) & \text{nếu } a > b^d \end{cases}$$

Từ $n = b^k$, ta có $k = \log_b n$. Do đó,

■ Nếu $a = b^d$, ta có $f(n) = O(a^{\log_b n}) + O(n^d \log_b n) = O(n^{\log_b a}) + O(n^d \log_b n) = O(n^d \log n)$. (Chú ý là khi $a \leq b^d$, ta có $n^{\log_b a} \leq n^d$)

■ Nếu $a < b^d$, ta có $f(n) = O(n^d) + O(n^d) = O(n^d)$. (Chú ý là khi $a \leq b^d$, ta có $n^{\log_b a} \leq n^d$)

■ Nếu $a > b^d$, ta có

$$f(n) = O(a^{\log_b n}) + O(a^{\log_b n}) = O(a^{\log_b n}) = O(n^{\log_b a}).$$

$$\begin{aligned} \text{(Chú ý là khi } a > b^d, \text{ ta có } (a/b^d)^{\log_b n} &= (b^{\log_b(a/b^d)})^{\log_b n} = \\ &= (b^{\log_b a - d})^{\log_b n} = n^{\log_b(a/b^d)} = n^{\log_b a - d} = O(n^{\log_b a}) \end{aligned}$$

5 Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)



Định lý 4: Định lý thợ

Gọi f là một hàm tăng thỏa mãn hệ thức truy hồi

$$f(n) = af(n/b) + g(n)$$

trong đó $a \geq 1$, $b > 1$ là các hằng số nào đó, và $g(n)$ là một hàm thỏa mãn $g(n) \geq 0$ khi n đủ lớn. Ta có

- (1) $f(n)$ là $\Theta(g(n))$ nếu $g(n) = \Omega(n^{\log_a b + \epsilon})$ với $\epsilon > 0$ và $g(n)$ thỏa mãn điều kiện $ag(n/b) \leq cg(n)$ với n đủ lớn và $c < 1$ là hằng số dương nào đó
- (2) $f(n)$ là $\Theta(n^{\log_b a} \log^{k+1} n)$ nếu $g(n) = \Theta(n^{\log_b a} \log^k n)$ với $k \geq 0$ nào đó. (Trong phần lớn các trường hợp thì $k = 0$.)
- (3) $f(n)$ là $\Theta(n^{\log_b a})$ nếu $g(n) = O(n^{\log_b a - \epsilon})$ với $\epsilon > 0$

6 Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Có thể bỏ qua sàn và trần



Thuật toán II

Hoàng Anh Đức

Định lý thợ

7 Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Chú ý

Có thể bỏ qua sàn và trần khi phân tích độ phức tạp của thuật toán

Ví dụ 14 (Thời gian chạy của thuật toán sắp xếp trộn)

- Nếu bạn phân tích giả mã của thuật toán sắp xếp trộn (Thuật toán 7), bạn sẽ nhận ra rằng thời gian chạy của thuật toán tuân theo hệ thức:

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + O(n) \quad (1)$$

(Xem Bài tập 6)

- Tuy nhiên, trong nhiều tài liệu, bạn sẽ thấy đề cập rằng thời gian chạy của thuật toán tuân theo hệ thức

$$T(n) = 2T(n/2) + O(n) \quad (2)$$

Có thể bỏ qua sàn và trần



Thuật toán II

Hoàng Anh Đức

Định lý thợ

8

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 14 (Thời gian chạy của thuật toán sắp xếp trộn)

Cả hai hệ thức trên đều đúng. Lý do ở đây là ta có thể bỏ qua sàn và trần bằng cách lý luận như sau:

- Đầu tiên, do chúng ta chỉ quan tâm đến chặn trên (vì ta luôn đánh giá theo O -lớn), ta có thể đánh giá

$$T(n) \leq 2T(\lceil n/2 \rceil) + O(n) \leq 2T(n/2 + 1) + O(n)$$

- Tiếp đó, ta định nghĩa một hàm mới $S(n) = T(n + \alpha)$ bằng cách chọn giá trị của α sao cho $S(n) \leq 2S(n/2) + O(n)$. Ta có

$$S(n) = T(n + \alpha)$$

định nghĩa của S

$$\leq 2T(n/2 + \alpha/2 + 1) + O(n + \alpha)$$

đánh giá cho T

$$= 2S(n/2 - \alpha/2 + 1) + O(n + \alpha)$$

định nghĩa của S

Chọn $\alpha = 2$, ta có $S(n) \leq 2S(n/2) + O(n)$

- Ước lượng $S(n)$ (ví dụ, bằng Định lý thợ), ta có $S(n) = O(n \log n)$, và do đó $T(n - 2) = O((n - 2) \log(n - 2)) = O(n \log n)$

Có thể bỏ qua sàn và trần



Thuật toán II
Hoàng Anh Đức

Bài tập 11

Xấp xỉ hệ thức truy hồi $T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + n - 1$ với điều kiện ban đầu $T(2) = 1$ và $T(1) = 0$. (**Chú ý:** $T(n)$ là thời gian chạy của thuật toán sắp xếp trộn)

(**Gợi ý:** Với $n = 2^k$ ($k \in \mathbb{N}$), chứng minh $T(n) = n \log n - n + 1$. Sử dụng quy nạp, chứng minh với mọi $n \in \mathbb{Z}^+$, $T(n) \approx n \log n - n + 1$.)

Bài tập 12

Cây van Emde Boas (van Emde Boas tree, vEB tree, hoặc van Emde Boas priority queue) là một cấu trúc dữ liệu đệ quy cho phép chúng ta chèn, xóa và tìm kiếm các khóa từ một tập hợp $U = \{1, 2, \dots, n\}$ một cách nhanh chóng. (Nó giải quyết cùng một vấn đề như cây tìm kiếm nhị phân, nhưng thời gian chạy sẽ tính theo kích thước của tập vũ trụ U thay vì số lượng khóa được lưu trữ.) Cây van Emde Boas có thời gian chạy cho bởi $T(n) = T(\sqrt{n}) + 1$ và $T(1) = 1$. Hãy giải hệ thức truy hồi này. (**Gợi ý:** Đặt $S(k) = T(2^k)$). **Đáp án:** $T(n) = O(\log \log n)$)

Bài tập 13

Ước lượng các hệ thức truy hồi sau:

(a) $T(n) = T(\lceil n/3 \rceil) + T(\lfloor 2n/3 \rfloor) + O(n)$

(b) $(\star) T(n) = \sqrt{n}T(\lceil \sqrt{n} \rceil) + O(n)$ (**Gợi ý:** Đặt $S(k) = T(2^{2^k})$). **Đáp án:**
 $T(n) = O(n \log \log n)$)

Định lý thợ

9 Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Có thể bỏ qua sàn và trần



Thuật toán II
Hoàng Anh Đức

Định lý thợ

10 Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Bài tập 14 (★)

Cho một mảng đã sắp xếp $A[1 \dots n]$, và ta cần tìm chỉ số i thỏa mãn $A[i-1] < x \leq A[i]$. (Bài toán này có thể giải được bằng thuật toán tìm kiếm nhị phân.) Dưới đây là mô tả thuật toán ROOTSEARCH để giải quyết bài toán này:

- Nếu n nhỏ (giả sử, nhỏ hơn 100), tìm chỉ số bằng thuật toán tìm kiếm tuyến tính. Ngược lại:
- Định nghĩa mileposts $:= A[\sqrt{n}], A[2\sqrt{n}], A[3\sqrt{n}], \dots, A[n]$ là danh sách các phần tử cách đều nhau $\sqrt{n}$ phần tử trong A .
- Đệ quy, tìm post $:= \text{ROOTSEARCH}(\text{mileposts}, x)$.
- Trả về $\text{ROOTSEARCH}(A[(\text{post} - 1)\sqrt{n}, \dots, \text{post}\sqrt{n}], x)$.

Có thể bỏ qua sàn và trần



Bài tập 14 (tiếp tục)

Thuật toán II

Hoàng Anh Đức

Định lý thợ

11 Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

ROOTSEARCH($A[1 \dots n], x$):

if $n < 100$ then *«Trường hợp cơ sở: mảng nhỏ»*

«Tìm kiếm tuyến tính»

for $i := 1$ to n do

if $A[i - 1] < x \leq A[i]$ then

return i

«Định nghĩa các cột mốc»

$k := \lfloor \sqrt{n} \rfloor$

mileposts := new array of size $\lceil n/k \rceil$

for $i := 1$ to $\lceil n/k \rceil$ do

mileposts[i] := $A[\min(i \cdot k, n)]$

«Tìm cột mốc thích hợp»

post := ROOTSEARCH(mileposts, x)

«Tìm kiếm trong đoạn con đã xác định»

return ROOTSEARCH($A[(post - 1) \cdot k + 1 \dots \min(post \cdot k, n)], x$)

Hãy xác định hệ thức truy hồi biểu diễn thời gian chạy của thuật toán ROOTSEARCH và ước lượng hệ thức bạn tìm được

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

12

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Bài toán: Tính số Fibonacci thứ n , được định nghĩa như sau:

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2} \quad \text{với } n \geq 2$$

Thuật toán 9: Thuật toán tính số Fibonacci thứ n

Input: Số nguyên không âm n

Output: Số Fibonacci thứ n

procedure Fibonacci(n):

if $n = 0$ **then**

return 0

else

if $n = 1$ **then**

return 1

else

return Fibonacci($n - 1$) +
 Fibonacci($n - 2$)

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Ví dụ 15 (Fibonacci(4))

Fibonacci(4)

Tính F_4 bằng đệ quy

Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sân và trần

Một số thuật toán đệ quy khác

13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

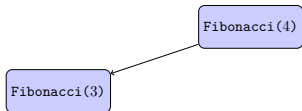
13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 15 (Fibonacci(4))



Đầu tiên, tính Fibonacci(3)

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

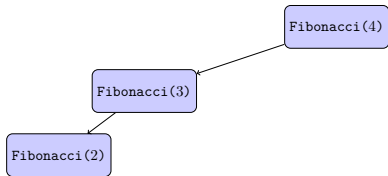
13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 15 (Fibonacci(4))



Để tính Fibonacci(3), ta cần tính Fibonacci(2)

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

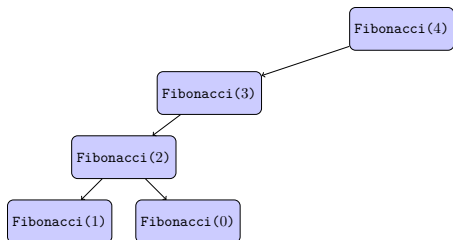
13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 15 (Fibonacci(4))



Để tính Fibonacci(2), ta cần Fibonacci(1) và Fibonacci(0)

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và
trần

Một số thuật toán đệ
quy khác

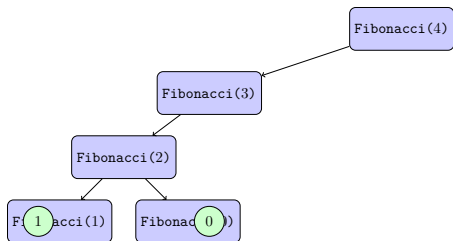
13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 15 (Fibonacci(4))



Trường hợp cơ sở: $\text{Fibonacci}(1) = 1$ và $\text{Fibonacci}(0) = 0$

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

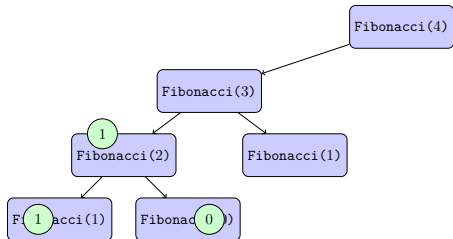
13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 15 (Fibonacci(4))



$$\text{Fibonacci}(2) = \text{Fibonacci}(1) + \text{Fibonacci}(0) = 1 + 0 = 1$$

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

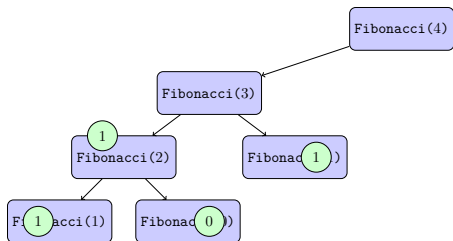
13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 15 (Fibonacci(4))



Với Fibonacci(3), ta cũng cần tính $\text{Fibonacci}(1) = 1$

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

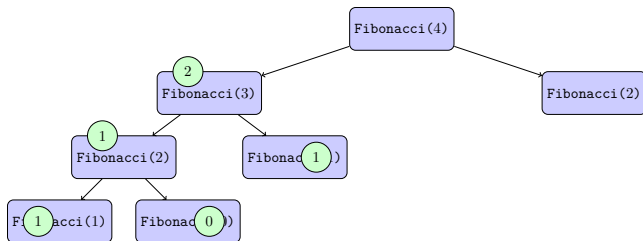
Một số thuật toán đệ quy khác

13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)



$$\text{Fibonacci}(3) = \text{Fibonacci}(2) + \text{Fibonacci}(1) = 1 + 1 = 2$$

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

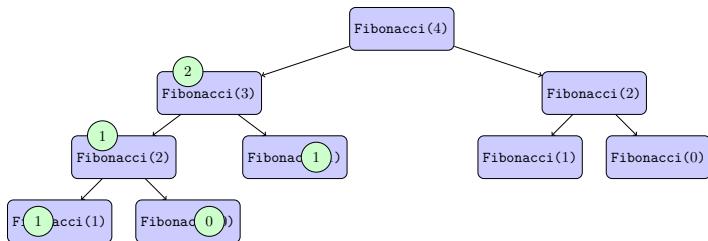
13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 15 (Fibonacci(4))



Bây giờ cho phần thứ hai của Fibonacci(4), tính Fibonacci(2)

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

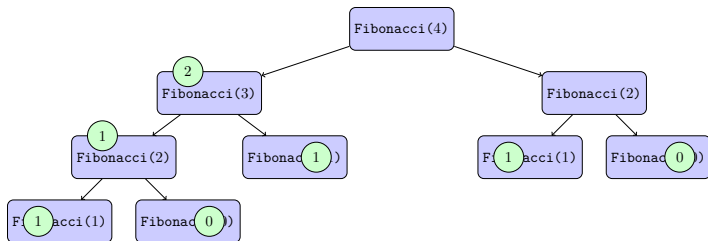
13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 15 (Fibonacci(4))



$$\text{Fibonacci}(2) = \text{Fibonacci}(1) + \text{Fibonacci}(0) = 1 + 0 = 1$$

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

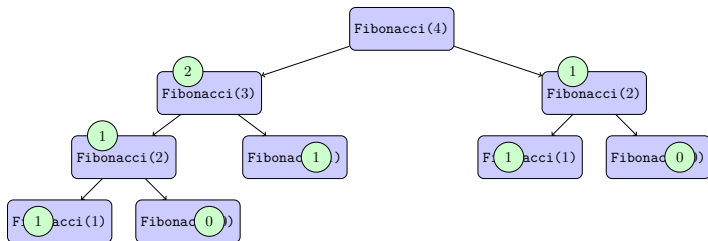
13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 15 (Fibonacci(4))



Cuối cùng, $\text{Fibonacci}(4) = \text{Fibonacci}(3) + \text{Fibonacci}(2) = 2 + 1 = 3$

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

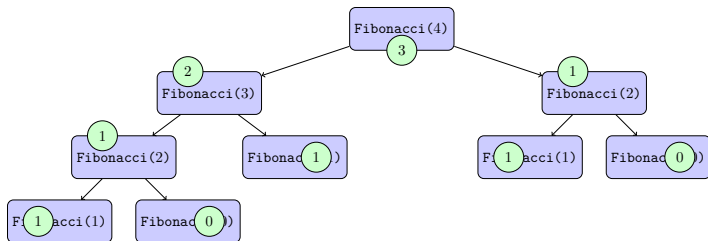
13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 15 (Fibonacci(4))



Cuối cùng, $\text{Fibonacci}(4) = \text{Fibonacci}(3) + \text{Fibonacci}(2) = 2 + 1 = 3$

Một số thuật toán đệ quy khác

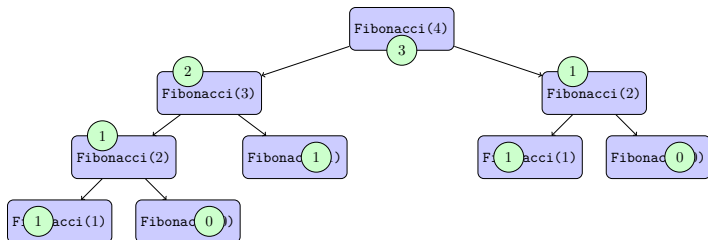
Tìm số Fibonacci thứ n



Thuật toán II

Hoàng Anh Đức

Ví dụ 15 (Fibonacci(4))



Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

13

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Bài tập 15

- (a) Chứng minh tính đúng đắn của thuật toán tìm số Fibonacci đệ quy
- (b) Chứng minh rằng thời gian chạy của thuật toán là $O(2^n)$
- (c) (★) Cải tiến thuật toán để giảm độ phức tạp thời gian

19

Một số thuật toán đệ quy khác

Tháp Hà Nội



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Bài toán: Di chuyển n đĩa từ cọc A sang cọc C với sự trợ giúp của cọc B, trong đó không có hai đĩa nào có cùng kích thước, tuân theo các quy tắc:

- Chỉ được di chuyển một đĩa mỗi lượt
- Không được đặt đĩa lớn hơn lên trên đĩa nhỏ hơn

Thuật toán 10: Thuật toán giải bài toán Tháp Hà Nội

Input: Số nguyên n , cọc nguồn A, cọc đích C, cọc trung gian B

Output: Dãy các bước di chuyển

procedure Hanoi(n, A, C, B):

if $n = 1$ **then**

 Di chuyển đĩa trên cùng từ A sang C

return

Hanoi($n - 1, A, B, C$)

Di chuyển đĩa trên cùng từ A sang C

Hanoi($n - 1, B, C, A$)

14

19

Một số thuật toán đệ quy khác

Tháp Hà Nội



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

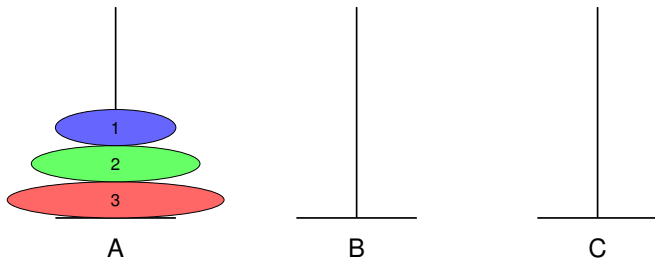
Tìm số Fibonacci thứ n

15 Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 16 (Minh họa giải thuật Tháp Hà Nội với 3 đĩa)

Trạng thái ban đầu: tất cả đĩa ở cọc A



Bài tập 16

- Chứng minh tính đúng đắn của thuật toán giải bài toán Tháp Hà Nội
- Chứng minh rằng số bước di chuyển tối thiểu để giải quyết bài toán với n đĩa là $2^n - 1$

Một số thuật toán đệ quy khác

Tháp Hà Nội



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

15

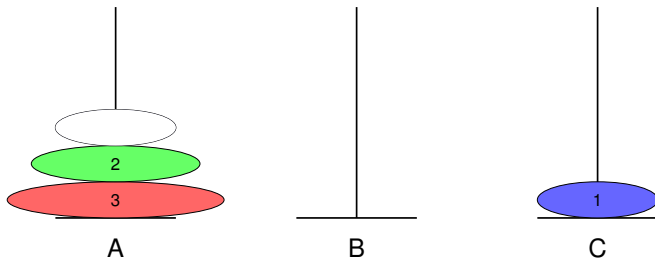
Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 16 (Minh họa giải thuật Tháp Hà Nội với 3 đĩa)

Bước 1: Di chuyển đĩa 1 từ A sang C

Hanoi(1, A, C, B)



Bài tập 16

- Chứng minh tính đúng đắn của thuật toán giải bài toán Tháp Hà Nội
- Chứng minh rằng số bước di chuyển tối thiểu để giải quyết bài toán với n đĩa là $2^n - 1$

Một số thuật toán đệ quy khác

Tháp Hà Nội



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

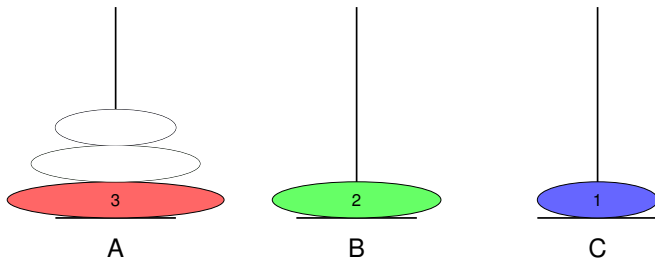
15 Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 16 (Minh họa giải thuật Tháp Hà Nội với 3 đĩa)

Bước 2: Di chuyển đĩa 2 từ A sang B

Đang thực hiện `Hanoi(2, A, B, C)`



Bài tập 16

- Chứng minh tính đúng đắn của thuật toán giải bài toán Tháp Hà Nội
- Chứng minh rằng số bước di chuyển tối thiểu để giải quyết bài toán với n đĩa là $2^n - 1$

Một số thuật toán đệ quy khác

Tháp Hà Nội



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

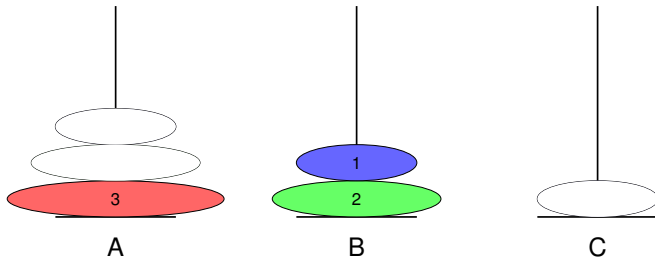
Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 16 (Minh họa giải thuật Tháp Hà Nội với 3 đĩa)

Bước 3: Di chuyển đĩa 1 từ C sang B

Hanoi(1, C, B, A)



Bài tập 16

- Chứng minh tính đúng đắn của thuật toán giải bài toán Tháp Hà Nội
- Chứng minh rằng số bước di chuyển tối thiểu để giải quyết bài toán với n đĩa là $2^n - 1$

15

19

Một số thuật toán đệ quy khác

Tháp Hà Nội



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sân và trần

Một số thuật toán đệ quy khác

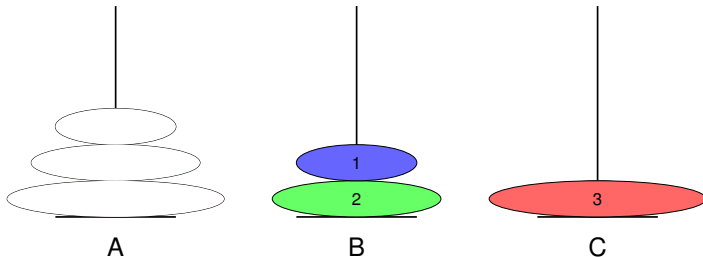
Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 16 (Minh họa giải thuật Tháp Hà Nội với 3 đĩa)

Bước 4: Di chuyển đĩa 3 từ A sang C



Bài tập 16

- Chứng minh tính đúng đắn của thuật toán giải bài toán Tháp Hà Nội
- Chứng minh rằng số bước di chuyển tối thiểu để giải quyết bài toán với n đĩa là $2^n - 1$

15

19

Một số thuật toán đệ quy khác

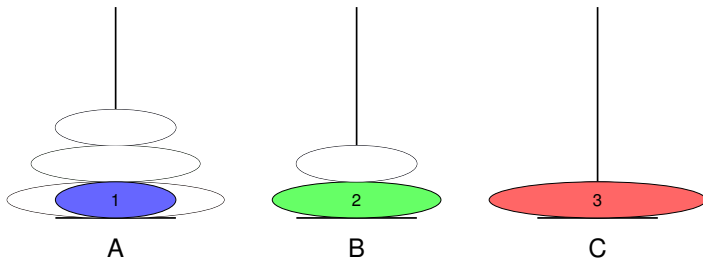
Tháp Hà Nội



Ví dụ 16 (Minh họa giải thuật Tháp Hà Nội với 3 đĩa)

Bước 5: Di chuyển đĩa 1 từ B sang A

Hanoi(1, B, A, C)



Bài tập 16

- Chứng minh tính đúng đắn của thuật toán giải bài toán Tháp Hà Nội
- Chứng minh rằng số bước di chuyển tối thiểu để giải quyết bài toán với n đĩa là $2^n - 1$

Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

15

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Một số thuật toán đệ quy khác

Tháp Hà Nội



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

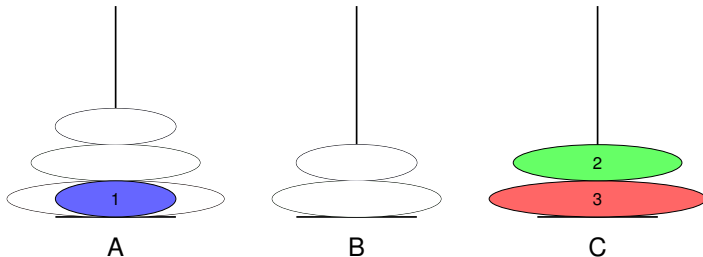
Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Ví dụ 16 (Minh họa giải thuật Tháp Hà Nội với 3 đĩa)

Bước 6: Di chuyển đĩa 2 từ B sang C

Đang thực hiện Hanoi (2, B, C, A)



Bài tập 16

- Chứng minh tính đúng đắn của thuật toán giải bài toán Tháp Hà Nội
- Chứng minh rằng số bước di chuyển tối thiểu để giải quyết bài toán với n đĩa là $2^n - 1$

15

19

Một số thuật toán đệ quy khác

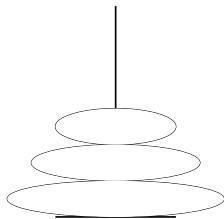
Tháp Hà Nội



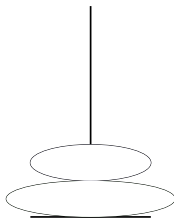
Ví dụ 16 (Minh họa giải thuật Tháp Hà Nội với 3 đĩa)

Bước 7: Di chuyển đĩa 1 từ A sang C

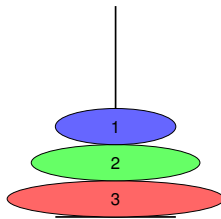
Hanoi(1, A, C, B)



A



B



C

Bài tập 16

- Chứng minh tính đúng đắn của thuật toán giải bài toán Tháp Hà Nội
- Chứng minh rằng số bước di chuyển tối thiểu để giải quyết bài toán với n đĩa là $2^n - 1$

Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

15

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Một số thuật toán đệ quy khác

Tháp Hà Nội



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

15

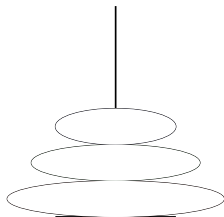
Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

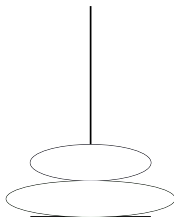
Ví dụ 16 (Minh họa giải thuật Tháp Hà Nội với 3 đĩa)

Hoàn thành! Tất cả các đĩa đã được chuyển từ cọc A sang cọc C

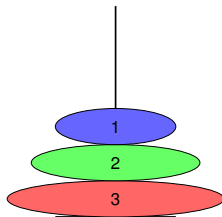
Total: $2^3 - 1 = 7$ bước



A



B



C

Bài tập 16

- Chứng minh tính đúng đắn của thuật toán giải bài toán Tháp Hà Nội
- Chứng minh rằng số bước di chuyển tối thiểu để giải quyết bài toán với n đĩa là $2^n - 1$

Một số thuật toán đệ quy khác

Sắp xếp nhanh (Quicksort)



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sàn và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n .

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Bài toán: Sắp xếp một mảng các phần tử theo thứ tự tăng dần.
Ý tưởng:

- Chọn một phần tử trong mảng làm *chốt (pivot)*
- Phân hoạch mảng thành hai phần: các phần tử nhỏ hơn pivot và các phần tử lớn hơn pivot
- Đặt pivot vào đúng vị trí của nó
- Đệ quy sắp xếp hai phần trước và sau pivot

16

19

Một số thuật toán đệ quy khác

Sắp xếp nhanh (Quicksort)



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Thuật toán 11: Thuật toán sắp xếp nhanh (Quicksort)

Input: Mảng A và các chỉ số p và r

Output: Mảng A được sắp xếp trong đoạn từ p đến r

```
1 procedure QuickSort( $A, p, r$ ):
2     // Sắp xếp mảng  $A$  từ vị trí  $p$  đến  $r$ 
3     if  $p < r$  then
4         // Điều kiện dừng: nếu mảng có ít nhất 2 phần tử
5          $q \leftarrow \text{Partition}(A, p, r)$            //  $q$  là vị trí cuối cùng của pivot
6         QuickSort( $A, p, q - 1$ )           // Đệ quy sắp xếp mảng con bên trái pivot
7         QuickSort( $A, q + 1, r$ )           // Đệ quy sắp xếp mảng con bên phải pivot
8
9 procedure Partition( $A, p, r$ ):
10    // Phân hoạch mảng và trả về vị trí của pivot
11     $x \leftarrow A[r]$            // Chọn pivot là phần tử cuối cùng của mảng
12     $i \leftarrow p - 1$            //  $i$  theo dõi vị trí cuối cùng của vùng nhỏ hơn pivot
13    for  $j \leftarrow p$  to  $r - 1$  do
14        // Duyệt qua tất cả các phần tử trừ pivot
15        if  $A[j] \leq x$  then
16            // Nếu phần tử hiện tại không lớn hơn pivot */
17             $i \leftarrow i + 1$            // Mở rộng vùng các phần tử nhỏ hơn pivot
18            Đổi chỗ  $A[i]$  và  $A[j]$            // Đưa phần tử nhỏ hơn pivot về đầu mảng
19
20    Đổi chỗ  $A[i + 1]$  và  $A[r]$            // Đặt pivot vào đúng vị trí của nó
21    return  $i + 1$            // Trả về vị trí của pivot trong mảng đã phân hoạch
```

Một số thuật toán đệ quy khác

Sắp xếp nhanh (Quicksort)



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và trần

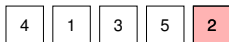
Một số thuật toán đệ quy khác

Tìm số Fibonacci thứ n

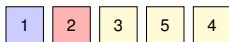
Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

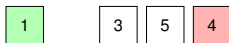
Ví dụ 17 (Minh họa thuật toán Quicksort)



Mảng ban đầu, pivot = 2

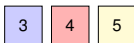


Sau phân hoạch: [1] pivot=2 [3,5,4]



Phần bên trái chỉ có 1 phần tử, đã sắp xếp

Phân hoạch phần bên phải, pivot = 4



Sau phân hoạch: [3] pivot=4 [5]



Phần bên phải đã sắp xếp



Mảng đã được sắp xếp hoàn chỉnh

18

19

Một số thuật toán đệ quy khác

Sắp xếp nhanh (Quicksort)



Thuật toán II

Hoàng Anh Đức

Định lý thợ

Có thể bỏ qua sản và
trần

Một số thuật toán đệ
quy khác

Tìm số Fibonacci thứ n .

Tháp Hà Nội

Sắp xếp nhanh (Quicksort)

Bài tập 17

- (a) Chứng minh tính đúng đắn của thuật toán sắp xếp nhanh
- (b) Phân tích độ phức tạp thời gian của thuật toán sắp xếp nhanh:
 - Trường hợp tốt nhất: $O(n \log n)$
 - Trường hợp xấu nhất: $O(n^2)$

19

19

1. Logic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Đệ quy
4. Thuật toán I: Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp
5. Thuật toán II: Thuật toán tham lam, thuật toán đệ quy
6. Lý thuyết số cơ bản
7. Các phương pháp đếm
8. Lý thuyết đồ thị I: Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông
9. Lý thuyết đồ thị II: Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị
10. Lý thuyết đồ thị III: Cây
11. Đại số Boole

VNU-HUS MAT3500: Toán rời rạc

Lý thuyết số cơ bản

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn



Nội dung



Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA



Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

- **Lý thuyết số (*number theory*)** nghiên cứu về các tính chất và mối quan hệ giữa các loại số
 - Đối tượng nghiên cứu quan trọng nhất là ***các số nguyên dương (*positive integers*)***
 - Đặc biệt chú trọng vào ***các số nguyên tố (*prime numbers*)*** và các tính chất của chúng
- Lý thuyết số là nền tảng cho nhiều ứng dụng quan trọng trong mật mã học hiện đại

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

3

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

- Cho các số nguyên a và b với $a \neq 0$. Ta nói b **chia hết cho** a , ký hiệu $b : a$, nếu tồn tại một số nguyên c sao cho $b = ac$.
- Trong trường hợp này, ta cũng nói a là **ước (factor)** của b hay b là **bội (multiple)** của a và ký hiệu $a \mid b$.
- Ta lần lượt sử dụng các ký hiệu $b \nmid a$ và $a \nmid b$ để chỉ b không chia hết cho a và a không là ước của b

Định lý 1

- (1) Nếu $a \mid b$ và $a \mid c$, thì $a \mid (b + c)$
- (2) Nếu $a \mid b$, thì $a \mid bc$
- (3) Nếu $a \mid b$ và $b \mid c$, thì $a \mid c$

Bài tập 1

Chứng minh Định lý 1

Bài tập 2

Chứng minh hoặc tìm phản ví dụ cho mệnh đề: “Với mọi số nguyên r, t, s, u , nếu r chia hết cho s và t chia hết cho u thì $r + s$ chia hết cho $t + u$ ”

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản



Định lý 2

Với $a \in \mathbb{Z}$ và $d \in \mathbb{Z}^+$, tồn tại duy nhất các số nguyên q và r , với $0 \leq r < d$, thỏa mãn $a = dq + r$

Chứng minh.

- Tồn tại các số nguyên q và r với $0 \leq r < d$ thỏa mãn $a = dq + r$
 - Chọn q là số nguyên lớn nhất thỏa mãn $dq \leq a$
 - Chọn $r = a - dq$. Ta có $0 \leq r < d$ (Tại sao?)
- Giả sử tồn tại các cặp số nguyên q_1, r_1 và q_2, r_2 thỏa mãn $a = dq_1 + r_1$ và $a = dq_2 + r_2$, với $0 \leq r_1 \leq r_2 < d$ và $(q_1, r_1) \neq (q_2, r_2)$
 - Nếu $q_1 = q_2$ thì $r_1 = a - dq_1 = a - dq_2 = r_2$
 - Do đó, $q_1 \neq q_2$. Theo giả thiết $a = dq_1 + r_1 = dq_2 + r_2$ và do đó $d = (r_2 - r_1)/(q_1 - q_2)$. Do $0 \leq r_1 \leq r_2 < d$, ta có $0 \leq r_2 - r_1 < d = (r_2 - r_1)/(q_1 - q_2)$. Do đó, $0 \leq q_1 - q_2 < 1$. Đây là một mâu thuẫn (Tại sao?)

Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

4 Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

5 Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân
Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố
Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

- Trong Định lý 2, a là **số bị chia (dividend)**, d là **số chia (divisor)**, q là **thương (quotient)**, và r là **số dư (remainder)**
- Ta cũng viết $q = a \operatorname{div} d$ và $r = a \bmod d$. Chú ý rằng với d cố định, $a \operatorname{div} d$ và $a \bmod d$ là các hàm từ $\mathbb{Z}$ đến $\mathbb{Z}$
- Ta có $q = \lfloor a/d \rfloor$ và $r = a - dq = a - d\lfloor a/d \rfloor$

Ví dụ 1

- $101 \operatorname{div} 11 = \quad$ và $101 \bmod 11 = \quad$
- $-101 \operatorname{div} 11 = \quad$ và $-101 \bmod 11 = \quad$
(Chú ý rằng mặc dù $-101 = 11(-9) - 2$ nhưng **số dư của phép chia $a = -101$ cho $d = 11$ không bằng -2** do $r = -2$ không thỏa mãn $0 \leq r < d$)

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

6 Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Thuật toán 1: Tìm thương và số dư

Input: $a \in \mathbb{Z}, d \in \mathbb{Z}^+$

Output: Thương q và số dư r của phép chia a cho d

```
1 procedure div-mod( $a, d$ ):  
2    $q := 0$   
3    $r := |a|$   
4   while  $r \geq d$  do // Tiếp tục trừ  $d$  từ  $r$  và tăng  $q$   
   cho đến khi  $r < d$   
5      $r := r - d$   
6      $q := q + 1$   
7   if  $a < 0$  và  $r > 0$  then // Trường hợp  $a$  âm  
8      $r := d - r$   
9      $q := -(q + 1)$   
10  return ( $q, r$ ) //  $q = a \text{ div } d$  là thương,  
    $r = a \bmod d$  là số dư
```

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

7

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giải thuật

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Ví dụ 2 (Thực hiện thuật toán tìm thương và số dư)

Với $(a, d) = (101, 11)$:

1. Khởi tạo: $q = 0, r = |101| = 101$
2. Vòng lặp **while**:
 - Khi $r = 101 \geq 11 = d$: $r = 101 - 11 = 90, q = 0 + 1 = 1$
 - Khi $r = 90 \geq 11 = d$: $r = 90 - 11 = 79, q = 1 + 1 = 2$
 - ... (tiếp tục 7 lần nữa)
 - Khi $r = 24 \geq 11 = d$: $r = 24 - 11 = 13, q = 8 + 1 = 9$
 - Khi $r = 13 \geq 11 = d$: $r = 13 - 11 = 2, q = 9 + 1 = 10$
3. Sau vòng lặp: $r = 2 < 11 = d$, dừng vòng lặp
4. Kiểm tra điều kiện **if**: $a = 101 > 0$ nên không thực hiện
5. Kết quả: $(q, r) = (9, 2)$

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

8 Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Ví dụ 3 (Thực hiện thuật toán tìm thương và số dư)

Với $(a, d) = (-101, 11)$:

1. Khởi tạo: $q = 0, r = |-101| = 101$
2. Vòng lặp **while**:
 - Khi $r = 101 \geq 11 = d$: $r = 101 - 11 = 90, q = 0 + 1 = 1$
 - Khi $r = 90 \geq 11 = d$: $r = 90 - 11 = 79, q = 1 + 1 = 2$
 - ... (tiếp tục 7 lần nữa)
 - Khi $r = 24 \geq 11 = d$: $r = 24 - 11 = 13, q = 8 + 1 = 9$
 - Khi $r = 13 \geq 11 = d$: $r = 13 - 11 = 2, q = 9 + 1 = 10$
3. Sau vòng lặp: $r = 2 < 11 = d$, dừng vòng lặp
4. Kiểm tra điều kiện **if**: $a = -101 < 0$ và $r = 2 > 0$ nên thực hiện
 - $r = 11 - 2 = 9$
 - $q = -(10 + 1) = -11$
5. Kết quả: $(q, r) = (-10, 9)$

Tính chia hết và phép toán môđun

Đồng dư theo môđun m



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

- Với $a, b \in \mathbb{Z}$ và $m \in \mathbb{Z}^+$, a đồng dư với b (theo) môđun m , ký hiệu $a \equiv b \pmod{m}$, khi và chỉ khi $m \mid (a - b)$

Định lý 3

Với $a, b \in \mathbb{Z}$ và $m \in \mathbb{Z}^+$, $a \equiv b \pmod{m}$ khi và chỉ khi $a \bmod m = b \bmod m$

Chứng minh.

($\Rightarrow$) Giả sử $a \equiv b \pmod{m}$. Giả sử $a = q_1m + r_1$ và $b = q_2m + r_2$ với $q_1, q_2 \in \mathbb{Z}$, $0 \leq r_1 < m$, và $0 \leq r_2 < m$. Ta chứng minh $a \bmod m = r_1 = r_2 = b \bmod m$

- Do $a \equiv b \pmod{m}$, ta có $m \mid (a - b)$
- Suy ra $m \mid ((q_1 - q_2)m + (r_1 - r_2))$. Do đó $m \mid (r_1 - r_2)$, nghĩa là $r_1 - r_2 = mp$ với $p \in \mathbb{Z}$
- Do $0 \leq r_1, r_2 < m$ nên $-m < r_1 - r_2 < m$
- Suy ra $-m < mp < m$ và do đó $p = 0$, nghĩa là $r_1 = r_2$

($\Leftarrow$) Giả sử $a \bmod m = b \bmod m = r$. Suy ra $a = q_1m + r$ và $b = q_2m + r$ với $q_1, q_2 \in \mathbb{Z}$. Do đó, $a - b = (q_1 - q_2)m$, nghĩa là $m \mid (a - b)$

Tính chia hết và phép toán môđun

Đồng dư theo môđun m



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

10 Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Bài tập 3

Chứng minh rằng quan hệ **đồng dư theo môđun m** “ $\equiv \pmod{m}$ ” là một quan hệ tương đương trên tập các số nguyên

Định lý 4

Với $a, b \in \mathbb{Z}$ và $m \in \mathbb{Z}^+$, $a \equiv b \pmod{m}$ khi và chỉ khi tồn tại $k \in \mathbb{Z}$ sao cho $a = b + km$

Chứng minh.

($\Rightarrow$) Giả sử $a \equiv b \pmod{m}$. Theo định nghĩa, $m \mid (a - b)$, nghĩa là tồn tại $k \in \mathbb{Z}$ sao cho $a - b = km$ hay $a = b + km$

($\Leftarrow$) Giả sử tồn tại $k \in \mathbb{Z}$ sao cho $a = b + km$. Suy ra $a - b = km$ và do đó $m \mid (a - b)$. Theo định nghĩa, $a \equiv b \pmod{m}$



Tính chia hết và phép toán môđun

Đồng dư theo môđun m



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

11 Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Định lý 5

Với $a, b, c, d \in \mathbb{Z}$ và $m \in \mathbb{Z}^+$, nếu $a \equiv b \pmod{m}$ và $c \equiv d \pmod{m}$ thì $a + c \equiv b + d \pmod{m}$ và $ac \equiv bd \pmod{m}$

Chứng minh.

Giả sử $a \equiv b \pmod{m}$ và $c \equiv d \pmod{m}$. Theo Định lý 4, tồn tại $s, t \in \mathbb{Z}$ thỏa mãn $a = b + sm$ và $c = d + tm$. Do đó,
 $a + c = (b + d) + (s + t)m$ và
 $ac = (b + sm)(d + tm) = bd + (bt + sd + stm)m$. Theo Định lý 4,
 $a + c \equiv b + d \pmod{m}$ và $ac \equiv bd \pmod{m}$ □

Hệ quả 6

- $(a + b) \bmod m = ((a \bmod m) + (b \bmod m)) \bmod m$
- $ab \bmod m = ((a \bmod m)(b \bmod m)) \bmod m$

Tính chia hết và phép toán môđun

Đồng dư theo môđun m



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

12 Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Bài tập 4

Chứng minh rằng nếu $a \equiv b \pmod{m}$ và $c \equiv d \pmod{m}$, trong đó $a, b, c, d \in \mathbb{Z}$ và $m \in \mathbb{Z}$ thỏa mãn $m \geq 2$, thì $a - c \equiv b - d \pmod{m}$

Bài tập 5

Tính các biểu thức sau

- (a) $(-133 \bmod 23 + 261 \bmod 23) \bmod 23$
- (b) $((457 \bmod 23) \cdot (182 \bmod 23)) \bmod 23$
- (c) $(99^2 \bmod 32)^3 \bmod 15$
- (d) $(3^4 \bmod 17)^2 \bmod 11$

Bài tập 6

Chứng minh rằng tích của ba số nguyên liên tiếp bất kỳ chia hết cho 6

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

- Thông thường, chúng ta biểu diễn các số theo **hệ cơ số (base) 10**, sử dụng các **chữ số (digit)** từ 0 đến 9
- Ta có thể **biểu diễn các số theo hệ cơ số $b > 1$ bất kỳ**
- Với mọi $n, b \in \mathbb{Z}^+$ ($b > 1$), tồn tại duy nhất một dãy $a_k a_{k-1} \dots a_1 a_0$ gồm các **chữ số** $a_i < b$ ($0 \leq i \leq k$) thỏa mãn

$$n = a_k b^k + a_{k-1} b^{k-1} + a_{k-2} b^{k-2} + \dots + a_1 b^1 + a_0 = \sum_{i=0}^k a_i b^i$$

13

- Ta cũng ký hiệu $n = (a_k a_{k-1} \dots a_2 a_1 a_0)_b$
- Một số hệ cơ số phổ biến

- **Hệ cơ số 10 (hệ thập phân (decimal))**: sử dụng 10 chữ số 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (do chúng ta có 10 ngón tay)
- **Hệ cơ số 2 (nhị phân (binary))**: sử dụng 2 chữ số 0, 1 (dùng trong tất cả các hệ thống máy tính hiện đại)
- **Hệ cơ số 8 (hệ bát phân (octal))**: sử dụng 8 chữ số 0, 1, 2, 3, 4, 5, 6, 7 (tương ứng với các nhóm 3 bit)
- **Hệ cơ số 16 (hệ thập lục phân (hexadecimal))**: sử dụng 16 chữ số 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (tương ứng với các nhóm 4 bit)

Biểu diễn số nguyên

Biểu diễn theo hệ *b*-phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo m

Biểu diễn số nguyên

Biểu diễn theo hệ *b*-phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Ví dụ 4 (Phân quyền tệp tin (file permission))

Ký hiệu số trong hệ bát phân sử dụng để biểu diễn *phân quyền tệp tin trong các hệ điều hành Linux*

Thông tin tệp tin trong Linux

`-rw-r--r-- 1 hoanganhduc hoanganhduc 1085121 Oct 21 15:17 Basic_Number_Theory.pdf`

Thông tin thư mục trong Linux

`drwxr-xr-x 3 hoanganhduc hoanganhduc 4096 Oct 21 15:24 Basic_Number_Theory`

Loại	Phân quyền owner	Phân quyền group	Phân quyền public
-	r w -	r - -	r - -
d	r w x	r - x	r - x

Ký hiệu số tương ứng của các quyền r (read, đọc), w (write, ghi), x (execute, thực thi)								
Phân quyền owner			Phân quyền group			Phân quyền public		
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

`rw-r--r--` $\longleftrightarrow$ 644

`rwXr-xr-x` $\longleftrightarrow$ 755

14

75

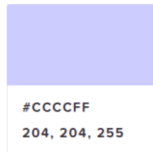
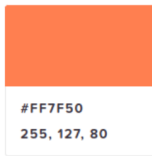
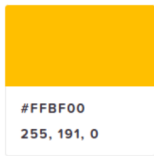
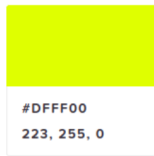
Biểu diễn số nguyên

Biểu diễn theo hệ b -phân



Ví dụ 5 (Mã màu)

Các số trong hệ thập lục phân được sử dụng để biểu diễn **mã màu (color code)** (nhằm đảm bảo các màu sắc được sử dụng một cách chính xác)



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Hình: Một số mã màu từ trang <https://htmlcolorcodes.com/>

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

16 Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Chuyển số từ hệ b -phân sang hệ thập phân ($b > 1$)

$$(a_k a_{k-1} \dots a_1 a_0)_b = \sum_{i=0}^k a_i b^i$$

$$\blacksquare (101011111)_2 = 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 351$$

$$\blacksquare (2AE0B)_{16} = 2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16^1 + 11 \cdot 16^0 = 175627$$

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Chuyển số từ hệ thập phân sang hệ b -phân ($b > 1$)

- (1) Tìm giá trị của chữ số ngoài cùng bên phải bằng cách tính $n \bmod b$
- (2) Gán $n := n \operatorname{div} b$
- (3) Lặp lại các bước (1) và (2) cho đến khi $n = 0$

$$\begin{aligned}n &= bq_0 + a_0 & n &:= q_0 \\&= b(bq_1 + a_1) + a_0 \\&= b^2q_1 + ba_1 + a_0 \\&\vdots \\&= b^k(0 + a_k) + b^{k-1}a_{k-1} + \dots b^3a_3 + b^2a_2 + ba_1 + a_0 & n &:= 0 \\&= b^ka_k + b^{k-1}a_{k-1} + \dots b^3a_3 + b^2a_2 + ba_1 + a_0\end{aligned}$$

Bài tập 7

Mô tả thuật toán trên bằng giả mã

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân



Ví dụ 6

$$(12345)_{10} = (?)_8$$

$$12345 = 8 \cdot 1543 + 1$$

$$1543 = 8 \cdot 192 + 7$$

$$192 = 8 \cdot 24 + 0$$

$$24 = 8 \cdot 3 + 0$$

$$3 = 8 \cdot 0 + 3$$

Do đó, $(12345)_{10} = (30071)_8$

Bài tập 8

(a) $(177130)_{10} = (?)_2$

(b) $(177130)_{10} = (?)_8$

(c) $(177130)_{10} = (?)_{16}$

Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

18 Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Biểu diễn số nguyên

Chuyển đổi giữa các hệ nhị phân, bát phân, và thập lục phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo m

Biểu diễn số nguyên

19 Biểu diễn theo hệ b -phân
Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Chuyển đổi giữa hệ nhị phân và bát/thập lục phân

- Mỗi chữ số trong hệ bát phân tương ứng với một khối 3 bit trong biểu diễn nhị phân
- Mỗi chữ số trong hệ thập lục phân tương ứng với một khối 4 bit trong biểu diễn nhị phân

Thập phân	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Thập lục phân	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Bát phân	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Nhị phân	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

$$(11\ 1110\ 1011\ 1111)_2 = (3EBF)_{16}$$

$$(A3D)_{16} = (1010\ 0011\ 1101)_2$$

Biểu diễn số nguyên

Chuyển đổi giữa các hệ nhị phân, bát phân, và thập lục phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

20

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Bài tập 9

(a) $(11111010111100)_2 = (?)_8$

(b) $(11111010111100)_2 = (?)_{16}$

(c) $(765)_8 = (?)_2$

(d) $(A8D)_{16} = (?)_2$

Biểu diễn số nguyên

Cộng và nhân các số nhị phân



Để cộng hai số nhị phân $a = (a_{n-1}a_{n-2} \dots a_1a_0)_2$ và $b = (b_{n-1}b_{n-2} \dots b_1b_0)_2$

- Cộng hai chữ số nhị phân ngoài cùng bên phải

$$a_0 + b_0 = c_0 \cdot 2 + s_0,$$

trong đó s_0 là chữ số ngoài cùng bên phải trong biểu diễn nhị phân của tổng $a + b$ và **nhớ (carry)** c_0

- Cộng hai chữ số nhị phân tiếp theo và nhớ

$$a_1 + b_1 + c_0 = c_1 \cdot 2 + s_1,$$

trong đó s_1 là chữ số tiếp theo (tính từ bên phải) trong biểu diễn nhị phân của tổng $a + b$ và nhớ c_1

- Tiếp tục cộng hai chữ số nhị phân tiếp theo và nhớ để xác định chữ số tiếp theo (tính từ bên phải) trong biểu diễn nhị phân của tổng $a + b$ và nhớ

- Ở bước cuối cùng, tính

$$a_{n-1} + b_{n-1} + c_{n-2} = c_{n-1} \cdot 2 + s_{n-1},$$

và chữ số đầu tiên trong biểu diễn nhị phân của tổng $a + b$ là $s_n = c_{n-1}$

Thuật toán trên cho ta $a + b = (s_ns_{n-1} \dots s_1s_0)_2$

Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

21

75

Biểu diễn số nguyên

Cộng và nhân các số nhị phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

22

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Thuật toán 2: Cộng hai số nhị phân

Input: $a = (a_{n-1} \dots a_0)_2, b = (b_{n-1} \dots b_0)_2$: biểu diễn nhị phân của các số nguyên dương a, b

Output: $s = (s_n s_{n-1} \dots s_0)$: biểu diễn nhị phân của $s = a + b$

```
1 procedure add( $a, b$ ):  
2    $c := 0$   
3   for  $j := 0$  to  $n - 1$  do  
4      $d := \lfloor (a_j + b_j + c)/2 \rfloor$   
5      $s_j = a_j + b_j + c - 2d$   
6      $c := d$   
7    $s_n := c$   
8   return  $(s_0, s_1, \dots, s_n)$ 
```

Biểu diễn số nguyên

Cộng và nhân các số nhị phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

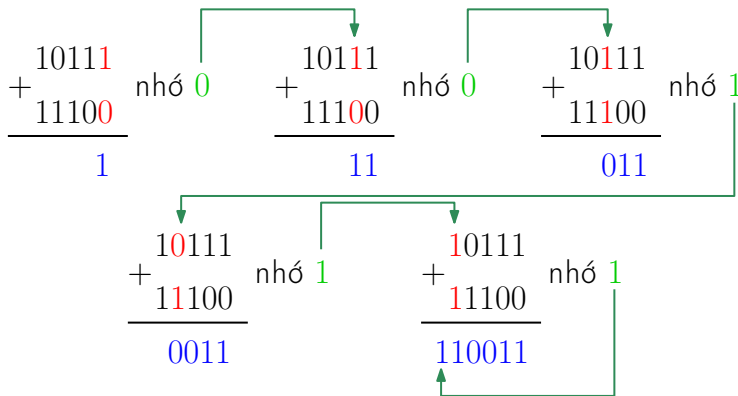
Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Ví dụ 7

Cộng hai số $a = (10111)_2$ và $b = (11100)_2$



23

75

Biểu diễn số nguyên

Cộng và nhân các số nhị phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

24

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Để nhân hai số nhị phân $a = (a_{n-1}a_{n-2} \dots a_1a_0)_2$ và $b = (b_{n-1}b_{n-2} \dots b_1b_0)_2$, chú ý rằng

$$\begin{aligned} ab &= a(b_02^0 + b_12^1 + \dots + b_{n-1}2^{n-1}) \\ &= a(b_02^0) + a(b_12^1) + \dots + a(b_{n-1}2^{n-1}) \end{aligned}$$

Phương trình này cho ta cách tính ab :

- Chú ý rằng $ab_j = a$ nếu $b_j = 1$ và $ab_j = 0$ nếu $b_j = 0$
- Mỗi lần nhân một số hạng với 2, ta dịch chuyển biểu diễn nhị phân của số đó sang trái một đơn vị và thêm 0 vào đuôi của biểu diễn. Nói cách khác, ta có thể thu được biểu diễn nhị phân của $(ab_j)2^j$ bằng cách dịch chuyển biểu diễn nhị phân của ab_j sang trái j đơn vị và thêm j số 0 vào đuôi của biểu diễn
- Cuối cùng, ta nhận được ab bằng cách cộng biểu diễn nhị phân của n số $(ab_j)2^j$ với $j \in \{0, \dots, n-1\}$

Biểu diễn số nguyên

Cộng và nhân các số nhị phân



Thuật toán 3: Nhân hai số nhị phân

Input: $a = (a_{n-1} \dots a_0)_2, b = (b_{n-1} \dots b_0)_2$: biểu diễn nhị phân của các số nguyên dương a, b

Output: biểu diễn nhị phân của $p = ab$

```
1 procedure multiply( $a, b$ ):  
2   for  $j := 0$  to  $n - 1$  do  
3     if  $b_j = 1$  then  
4        $c_j := a$  sau khi di chuyển  $j$  đơn vị sang trái  
5     else  
6        $c_j := 0$   
7     //  $c_0, \dots, c_{n-1}$  là các tích thành phần  
8      $p := 0$   
9     for  $j := 0$  to  $n - 1$  do  
10       $p := \text{add}(p, c_j)$   
11  return  $p$ 
```

Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

25

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Biểu diễn số nguyên

Cộng và nhân các số nhị phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Ví dụ 8

Nhân hai số $a = (110)_2$ và $b = (101)_2$

$$\begin{array}{r} 110 \\ \times 101 \\ \hline 110 \end{array}$$

$$\begin{array}{r} 110 \\ \times 101 \\ \hline 110 \\ 0000 \end{array}$$

$$\begin{array}{r} 110 \\ \times 101 \\ \hline 110 \\ + 0000 \\ \hline 11000 \\ + 1100 \\ \hline 11110 \end{array}$$

26

75

Biểu diễn số nguyên

Cộng và nhân các số nhị phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

27

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Bài tập 10

Tính tổng và tích các số nhị phân sau

(a) $(1000111)_2$ và $(1110111)_2$

■ **Kết quả:** Tổng = $(10111110)_2$, Tích = $(10000100000001)_2$

(b) $(11101111)_2$ và $(10111101)_2$

■ **Kết quả:** Tổng = $(110101100)_2$, Tích = $(1011000001110011)_2$

Biểu diễn số nguyên

Biểu diễn các số nguyên âm theo hệ nhị phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

28

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

- Trong thực hành, chúng ta cần biểu diễn không chỉ các số dương (positive integers) mà cả các số âm (negative integers)
- Khi sử dụng “giấy và bút”, các số âm được thể hiện bằng cách thêm dấu “-” đằng trước
- Khi sử dụng máy tính, tất cả các loại dữ liệu đều được biểu diễn trong hệ nhị phân

Biểu diễn số nguyên

Biểu diễn các số nguyên âm theo hệ nhị phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Biểu diễn thông qua *định dạng dấu-lượng (sign-magnitude format)*

- Bit ngoài cùng bên trái dùng để biểu diễn dấu (0 là dương, 1 là âm)
- Phần còn lại biểu diễn độ lớn (hay trị tuyệt đối) của số
- Một chuỗi nhị phân n bit có thể biểu diễn bất kỳ số nguyên i nào thỏa mãn $-(2^{n-1} - 1) \leq i \leq 2^{n-1} - 1$
- **Ví dụ:** biểu diễn nhị phân của 93 là 01011101 và biểu diễn nhị phân của -93 là 11011101
- **Hạn chế:** Số 0 có hai biểu diễn nhị phân: 000...00 (biểu diễn $+0$) và 100...00 (biểu diễn -0)

29

Biểu diễn số nguyên

Biểu diễn các số nguyên âm theo hệ nhị phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Biểu diễn thông qua *ký hiệu phần bù một (one's complement notation)*

- Bit ngoài cùng bên trái dùng để biểu diễn dấu (0 là dương, 1 là âm)
- Khi biểu diễn bằng ký hiệu phần bù một, nếu $+a = (a_{n-1} \dots a_0)_2$ thì $-a = (\overline{a_{n-1} \dots a_0})_2$, trong đó $\overline{a_{n-1} \dots a_0}$ là phần bù của $a_{n-1} \dots a_0$ thu được thông qua tính toán bằng toán tử logic $\neg$ (phủ định) theo từng bit
- Một chuỗi nhị phân n bit có thể biểu diễn bất kỳ số nguyên i nào thỏa mãn $-(2^{n-1} - 1) \leq i \leq 2^{n-1} - 1$
- **Ví dụ:** biểu diễn nhị phân của 93 là 01011101 và biểu diễn nhị phân của -93 là 10100010
- **Hạn chế:** Số 0 có hai biểu diễn nhị phân: 00...00 (biểu diễn +0) và 11...11 (biểu diễn -0)

30

75

Biểu diễn số nguyên

Biểu diễn các số nguyên âm theo hệ nhị phân



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Biểu diễn thông qua *ký hiệu phần bù hai (two's complement notation)*

- Bit ngoài cùng bên trái dùng để biểu diễn dấu (0 là dương, 1 là âm)
- Khi biểu diễn bằng ký hiệu phần bù hai, nếu $a = (a_{n-1} \dots a_0)_2$ thì $-a = (\overline{a_{n-1} \dots a_0})_2 + 1$, trong đó $\overline{a_{n-1} \dots a_0}$ là phần bù của $a_{n-1} \dots a_0$ thu được thông qua tính toán bằng toán tử logic $\neg$ (phủ định) theo từng bit
- Trong trường hợp này, một chuỗi nhị phân n bit có thể biểu diễn bất kỳ số nguyên i nào thỏa mãn $-2^{n-1} \leq i < 2^{n-1}$

Ví dụ 9 (Với $n = 3$)

Giá trị	Chuỗi 3-bit	Giá trị	Chuỗi 3-bit
3	011	-3	101
2	010	-2	110
1	001	-1	111
0	000	-4	100

31

75

Tính chia hết và phép toán môđun

Tính lũy thừa môđun



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

32

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

- Trong các thuật toán mã hóa hiện đại, một bài toán quan trọng là *tính $b^n \bmod m$ một cách hiệu quả* mà không cần sử dụng quá nhiều bộ nhớ, đặc biệt là *khi b, n, m là các số nguyên lớn*
- Việc tính b^n rồi tìm số dư khi chia nó cho m là không thực tế, do b^n có thể cực lớn và ta sẽ cần một lượng lớn bộ nhớ chỉ để lưu giá trị của b^n
- Ta có thể tính $b^n \bmod m$ bằng cách lần lượt tính $b^k \bmod m$ cho $k = 1, 2, \dots, n$, sử dụng tính chất $b^{k+1} \bmod m = b(b^k \bmod m) \bmod m$. Tuy nhiên, hướng tiếp cận này cũng không thực tế, do ta cần thực hiện $n - 1$ phép nhân các số nguyên và n có thể rất lớn
- Ta trình bày một hướng tiếp cận hiệu quả *dựa trên biểu diễn nhị phân của n*

Tính chia hết và phép toán môđun

Tính lũy thừa môđun



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

■ Chú ý rằng

Biểu diễn nhị phân của n

$$\begin{aligned} b^n &= b^{a_{k-1}2^{k-1} + a_{k-2}2^{k-2} + \dots + a_12^1 + a_02^0} \\ &= (b^{2^{k-1}})^{a_{k-1}} \times (b^{2^{k-2}})^{a_{k-2}} \times \dots \times (b^{2^1})^{a_1} \times (b^{2^0})^{a_0} \end{aligned}$$

■ Chúng ta có thể tính các giá trị b^{2^j} bằng cách *liên tục bình phương*

■ Sau đó ta chỉ cần nhân các giá trị này với nhau để tạo thành một tích thành phần, tùy thuộc vào a_j có bằng 1 hay không

■ Quan trọng là, sau mỗi bước nhân, để tăng tính hiệu quả và tiết kiệm bộ nhớ, ta *có thể lấy mod m của kết quả để tiếp tục thực hiện tính toán*

33

75

Tính chia hết và phép toán môđun

Tính lũy thừa môđun



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

34 Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Ví dụ 10

Ta tính $3^{644} \bmod 645$

$$644 = 1 \times 2^9 + 0 \times 2^8 + 1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 \\ + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

$$3^{644} = (3^{2^9})^1 \times (3^{2^8})^0 \times (3^{2^7})^1 \times (3^{2^6})^0 \times (3^{2^5})^0 \times (3^{2^4})^0 \\ \times (3^{2^3})^0 \times (3^{2^2})^1 \times (3^{2^1})^0$$

Tính chia hết và phép toán môđun

Tính lũy thừa môđun



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

35

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Ta tính các giá trị $3^{2^j} \bmod 645$ ($1 \leq j \leq 9$) bằng cách liên tục bình phương và lấy $\bmod 645$

$$3^{2^1} \bmod 645 = 9$$

$$3^{2^2} \bmod 645 = (3^{2^1})^2 \bmod 645 = (3^{2^1} \bmod 645)^2 \bmod 645 = 81$$

$$3^{2^3} \bmod 645 = (3^{2^2})^2 \bmod 645 = (3^{2^2} \bmod 645)^2 \bmod 645 = 111$$

$$3^{2^4} \bmod 645 = (3^{2^3})^2 \bmod 645 = (3^{2^3} \bmod 645)^2 \bmod 645 = 66$$

$$3^{2^5} \bmod 645 = (3^{2^4})^2 \bmod 645 = (3^{2^4} \bmod 645)^2 \bmod 645 = 486$$

$$3^{2^6} \bmod 645 = (3^{2^5})^2 \bmod 645 = (3^{2^5} \bmod 645)^2 \bmod 645 = 126$$

$$3^{2^7} \bmod 645 = (3^{2^6})^2 \bmod 645 = (3^{2^6} \bmod 645)^2 \bmod 645 = 396$$

$$3^{2^8} \bmod 645 = (3^{2^7})^2 \bmod 645 = (3^{2^7} \bmod 645)^2 \bmod 645 = 81$$

$$3^{2^9} \bmod 645 = (3^{2^8})^2 \bmod 645 = (3^{2^8} \bmod 645)^2 \bmod 645 = 111$$

Tính chia hết và phép toán môđun

Tính lũy thừa môđun



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân
Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

36

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Do đó,

$$\begin{aligned}3^{644} \bmod 645 &= (3^{2^2} \times 3^{2^7} \times 3^{2^9}) \bmod 645 \\&= (((3^{2^2} \bmod 645) \times (3^{2^7} \bmod 645)) \bmod 645) \\&\quad \times (3^{2^9} \bmod 645) \bmod 645 \\&= (((81 \times 396) \bmod 45) \times 111) \bmod 645 \\&= (471 \times 111) \bmod 645 \\&= 36\end{aligned}$$

Tính chia hết và phép toán môđun

Tính lũy thừa môđun



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

37 Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Thuật toán 4: Tính lũy thừa môđun nhanh

Input: b : số nguyên, $n = (a_{k-1}a_{k-2} \dots a_1a_0)_2$: biểu diễn nhị phân của số nguyên dương n , m : số nguyên dương

Output: $b^n \bmod m$

```
1  $x := 1$  // để lưu trữ kết quả
2  $b2i := b \bmod m$  //  $b^{2^i}$ , đầu tiên  $i = 0$ 
3 for  $i := 0$  to  $k - 1$  do // xét tất cả  $k$  bit của  $n$ 
4     if  $a_i = 1$  then
5          $x := (x \cdot b2i) \bmod m$ 
6      $b2i := (b2i \cdot b2i) \bmod m$  //  $b^{2^{i+1}} = (b^{2^i}) \cdot (b^{2^i})$ 
7 return  $x$ 
```

Bài tập 11

Sử dụng thuật toán tính $b^n \bmod m$ thông qua biểu diễn nhị phân của n đã mô tả ở trên để tính $7^{644} \bmod 645$

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

38

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

- Một số nguyên $p > 1$ là một **số nguyên tố (prime number)** nếu các ước số dương duy nhất của p là 1 và chính nó
 - Ví dụ: 2, 3, 5, 11, ...
- Các số nguyên lớn hơn 1 và không phải là số nguyên tố được gọi là các **hợp số (composite number)**

Bài tập 12

Chứng minh rằng nếu p là một số nguyên tố và $p \mid ab$ với $a, b \in \mathbb{Z}^+$ thì $p \mid a$ hoặc $p \mid b$. (**Gợi ý:** Giả sử $p \nmid a$, chứng minh $p \mid b$. Sử dụng Định lý Bézout (Định lý 12)) sẽ đề cập ở phần sau.) Phát biểu trên có đúng với p là hợp số hay không? Tại sao?

Bài tập 13

Sử dụng quy nạp, hãy chứng minh phát biểu tổng quát: nếu p là một số nguyên tố và $p \mid a_1 a_2 \dots a_n$, trong đó $a_i \in \mathbb{Z}$ với $1 \leq i \leq n$, thì $p \mid a_j$ với j nào đó ($1 \leq j \leq n$)

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

39

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Định lý 7: Định lý cơ bản của số học

Mọi số nguyên dương lớn hơn 1 có thể được viết một cách duy nhất dưới dạng một số nguyên tố hoặc một tích của các ước nguyên tố của nó theo thứ tự tăng dần

Gợi ý.

- Ta đã chứng minh bằng phương pháp quy nạp: nếu $n > 1$ là một số nguyên thì n có thể được biểu diễn dưới dạng tích của các số nguyên tố
- Để chỉ ra tính “duy nhất”, ta chứng minh bằng phản chứng: giả sử số nguyên dương $n > 1$ có thể được biểu diễn dưới dạng tích các số nguyên tố theo hai cách, ví dụ như $n = p_1 p_2 \dots p_s$ và $n = q_1 q_2 \dots q_t$, trong đó mỗi p_i ($1 \leq i \leq s$) và q_j ($1 \leq j \leq t$) là một số nguyên tố thỏa mãn $p_1 \leq p_2 \leq \dots \leq p_s$ và $q_1 \leq q_2 \leq \dots \leq q_t$. Sử dụng Bài tập 13 để chỉ ra mâu thuẫn

75

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Định lý 8

Nếu $n \in \mathbb{Z}^+$ là một hợp số, thì n có một ước nguyên tố nhỏ hơn hoặc bằng $\sqrt{n}$

Chứng minh.

- Theo giả thiết, $n \in \mathbb{Z}^+$ là hợp số, do đó n có một ước số a thỏa mãn $1 < a < n$. Do đó, tồn tại số nguyên $b > 1$ sao cho $n = ab$.
- Ta chứng minh $a \leq \sqrt{n}$ hoặc $b \leq \sqrt{n}$. Thật vậy, giả sử $a > \sqrt{n}$ và $b > \sqrt{n}$. Suy ra, $ab > \sqrt{n} \cdot \sqrt{n} = n$, mâu thuẫn với định nghĩa của a, b . Do đó $a \leq \sqrt{n}$ hoặc $b \leq \sqrt{n}$, nghĩa là, n có một ước số lớn hơn 1 và không vượt quá $\sqrt{n}$ (a hoặc b)
- Theo Định lý cơ bản của số học, ước số này là một số nguyên tố hoặc có một ước nguyên tố nhỏ hơn nó. Trong cả hai trường hợp, n có một ước nguyên tố nhỏ hơn hoặc bằng $\sqrt{n}$

40

75

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân
Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân
Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

41

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

- Mệnh đề phản đảo của Định lý 8: Một số nguyên $n > 1$ là số nguyên tố nếu nó không chia hết cho bất kỳ số nguyên tố nào nhỏ hơn hoặc bằng $\sqrt{n}$
- Tìm các số nguyên tố giữa 2 và n bằng *Sàng Eratosthenes* (*The Sieve of Eratosthenes*)
 - (1) Viết các số $2, \dots, n$ vào một danh sách. Gán $i := 2$
 - (2) Bỏ đi tất cả các bội của i trừ chính nó khỏi danh sách
 - (3) Gọi k là số nhỏ nhất hiện có trong danh sách thỏa mãn $k > i$. Gán $i := k$
 - (4) Nếu $i > \sqrt{n}$ thì dừng lại, ngược lại thì quay lại bước (2)
- Việc kiểm tra xem một số có phải là số nguyên tố hay không có thể được thực hiện trong thời gian đa thức [Agrawal, Kayal, and Saxena 2004] (đa thức của số bit sử dụng để mô tả số đầu vào)

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân
Cộng và nhân các số nhị phân

Biểu diễn các số nguyên
âm theo hệ nhị phân
Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Định lý 9

Có vô hạn số nguyên tố

Chứng minh (theo Euclid).

- Giả sử chỉ có hữu hạn các số nguyên tố $p_1, p_2, \dots, p_n$. Đặt $Q = p_1 p_2 \dots p_n + 1$
- Theo Định lý cơ bản của số học, (a) Q là một số nguyên tố hoặc (b) Q có thể được viết thành tích của ít nhất hai số nguyên tố
- **(a) đúng:** Do đó, Q là số nguyên tố. Theo định nghĩa, $Q \notin \{p_1, \dots, p_n\}$, mâu thuẫn với giả thiết toàn bộ các số nguyên tố là $p_1, \dots, p_n$
- **(b) đúng:** Do đó, tồn tại j thỏa mãn $p_j \mid Q$ với $1 \leq j \leq n$. Chú ý rằng $p_j \mid (p_1 p_2 \dots p_n)$, và do đó $p_j \mid (Q - p_1 p_2 \dots p_n)$, suy ra $p_j \mid 1$, mâu thuẫn với giả thiết p_j là số nguyên tố

42

75

Số nguyên tố và Ước chung lớn nhất

Ước chung lớn nhất



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phân dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

- Cho $a, b \in \mathbb{Z}$ và a, b không đồng thời bằng 0. **Ước chung lớn nhất (greatest common divisor)** của a và b , ký hiệu $\gcd(a, b)$, là số nguyên lớn nhất d thỏa mãn $d \mid a$ và $d \mid b$
- Các số nguyên a và b được gọi là **nguyên tố cùng nhau (relatively prime hoặc coprime)** khi và chỉ khi $\gcd(a, b) = 1$
- Một tập các số nguyên $\{a_1, a_2, a_3, \dots, a_n\}$ được gọi là **đôi một nguyên tố cùng nhau (pairwise relatively prime)** nếu mọi cặp a_i, a_j với $1 \leq i < j \leq n$ là nguyên tố cùng nhau
- Nếu các số nguyên dương a và b được phân tích thành tích các số nguyên tố

$$a = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n} \quad b = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n}$$

trong đó các số mũ là các số nguyên không âm (có thể bằng 0), thì

$$\gcd(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \dots p_n^{\min(a_n, b_n)}$$

43

75

Số nguyên tố và Ước chung lớn nhất

Bội chung nhỏ nhất và liên hệ với Ước chung lớn nhất



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

- Cho $a, b \in \mathbb{Z}^+$. **Bội chung nhỏ nhất (least common multiple)** của a và b , ký hiệu $\text{lcm}(a, b)$, là số nguyên nhỏ nhất d thỏa mãn $a \mid d$ và $b \mid d$
 - Tập các bội chung của a và b có ít nhất một phần tử ab
 - **Tính sắp thứ tự tốt:** Mọi tập con khác rỗng của $\mathbb{Z}^+$ có phần tử nhỏ nhất
- Nếu a và b được phân tích thành tích các số nguyên tố

$$a = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n} \quad b = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n}$$

trong đó các số mũ là các số nguyên không âm (có thể bằng 0), thì

$$\text{lcm}(a, b) = p_1^{\max(a_1, b_1)} p_2^{\max(a_2, b_2)} \dots p_n^{\max(a_n, b_n)}$$

Định lý 10

Với $a, b \in \mathbb{Z}^+$, $ab = \text{gcd}(a, b) \cdot \text{lcm}(a, b)$

Bài tập 15

Chứng minh Định lý 10

44

75

Số nguyên tố và Ước chung lớn nhất

Thuật toán Euclid



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Bổ đề 11

Cho $a = bq + r$ với a, b, q, r là các số nguyên. Ta có $\gcd(a, b) = \gcd(b, r)$. Do đó, ta cũng có $\gcd(a, b) = \gcd(b, (a \bmod b))$

Chứng minh.

- Gọi D_{ab} là tập các ước số chung của a và b , với các số nguyên a, b bất kỳ. Ta chứng minh $D_{ab} = D_{br}$
- $D_{ab} \subseteq D_{br}$: Giả sử $x \in D_{ab}$. Theo định nghĩa, $x \mid a$ và $x \mid b$. Theo Định lý 1, $x \mid (a - bq)$ và do đó $x \mid r$, suy ra $x \in D_{br}$
- $D_{br} \subseteq D_{ab}$: Giả sử $x \in D_{br}$. Theo định nghĩa, $x \mid b$ và $x \mid r$. Theo Định lý 1, $x \mid (bq + r)$ và do đó $x \mid a$, suy ra $x \in D_{ab}$
- Từ $D_{ab} = D_{br}$, ta có $\gcd(a, b) = \gcd(b, r)$



45

75

Số nguyên tố và Ước chung lớn nhất

Thuật toán Euclid



Thuật toán Euclid

Ý tưởng: Sử dụng đẳng thức $\gcd(a, b) = \gcd(b, (a \bmod b))$

Ví dụ 11 (Thuật toán Euclid)

Tìm $\gcd(372, 164)$

$$\begin{aligned}\gcd(372, 164) &= \gcd(164, 372 \bmod 164) = \gcd(164, 44) \\ &= \gcd(44, 164 \bmod 44) = \gcd(44, 32) \\ &= \gcd(32, 44 \bmod 32) = \gcd(32, 12) \\ &= \gcd(12, 32 \bmod 12) = \gcd(12, 8) \\ &= \gcd(8, 12 \bmod 8) = \gcd(8, 4) \\ &= \gcd(4, 8 \bmod 4) = \gcd(4, 0) \\ &= 4\end{aligned}$$

Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

46

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Số nguyên tố và Ước chung lớn nhất

Thuật toán Euclid



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Thuật toán 5: Thuật toán Euclid

Input: a, b : các số nguyên dương

Output: $\gcd(a, b)$

```
1  $x := a$ 
2  $y := b$ 
3 while  $y \neq 0$  do
4    $r := x \bmod y$ 
5    $x := y$ 
6    $y := r$ 
7 return  $x$ 
```

// $x = \gcd(a, b)$

47

75

Số nguyên tố và Ước chung lớn nhất

Thuật toán Euclid



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Bài tập 16

Sử dụng thuật toán Euclid để tìm

(a) $\gcd(12, 18)$

(b) $\gcd(111, 201)$

(c) $\gcd(1001, 1331)$

Bài tập 17

Chứng minh rằng nếu a, b, m là các số nguyên với $m \geq 2$ và $a \equiv b \pmod{m}$ thì $\gcd(a, m) = \gcd(b, m)$. (**Gợi ý:** Chứng minh tập các ước chung của a và m bằng với tập các ước chung của b và m .)

48

75

Số nguyên tố và Ước chung lớn nhất

Ước chung lớn nhất và tổ hợp tuyến tính



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Định lý 12: Định lý Bézout

Cho các số nguyên dương a, b . Tồn tại các số nguyên s, t sao cho $\gcd(a, b) = sa + tb$

- Các số nguyên s, t thỏa mãn Định lý Bézout được gọi là các **hệ số Bézout (Bézout's coefficients)** của a và b
- Phương trình $\gcd(a, b) = sa + tb$ được gọi là **đẳng thức Bézout (Bézout's identity)**

Chú ý:

- Chúng ta không trình bày chứng minh của Định lý Bézout
- Chúng ta sẽ đề cập hai phương pháp để tìm một tổ hợp tuyến tính của hai số nguyên bằng với ước chung lớn nhất của chúng (Trong phần này, ta luôn giả thiết các tổ hợp tuyến tính chỉ có hệ số nguyên)
 - (1) Đi ngược lại theo các phép chia của thuật toán Euclid
 - (2) Thuật toán Euclid mở rộng (The extended Euclidean algorithm)

49

75

Số nguyên tố và Ước chung lớn nhất

Ước chung lớn nhất và tổ hợp tuyến tính



Ví dụ 12

Biểu diễn $\gcd(252, 198) = 18$ dưới dạng tổ hợp tuyến tính của 252 và 198

■ Thuật toán Euclid sử dụng các phép chia như sau

■ $252 = 1 \cdot 198 + 54$

■ $198 = 3 \cdot 54 + 36$

■ $54 = 1 \cdot 36 + 18$

■ $36 = 2 \cdot 18 + 0$

■ Ta có

$$\begin{aligned} 18 &= 54 - 1 \cdot 36 \\ &= 54 - 1 \cdot (198 - 3 \cdot 54) \\ &= 4 \cdot 54 - 1 \cdot 198 \\ &= 4 \cdot (252 - 1 \cdot 198) - 1 \cdot 198 \\ &= 4 \cdot 252 - 5 \cdot 198 \end{aligned}$$

Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

50

75

Số nguyên tố và Ước chung lớn nhất

Ước chung lớn nhất và tổ hợp tuyến tính



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Thuật toán 6: Thuật toán Euclid mở rộng

Input: a, b : các số nguyên dương

Output: (d, s, t) : $d = \gcd(a, b)$ và s, t thỏa mãn $d = sa + tb$

procedure ExtEuclid(a, b):

if $b = 0$ **then**

return $(a, 1, 0)$

$(d_1, s_1, t_1) := \text{ExtEuclid}(b, a \bmod b)$

$d := d_1$

$s := t_1$

$t := s_1 - (a \text{ div } b) \cdot t_1$

return (d, s, t)

51

75

Số nguyên tố và Ước chung lớn nhất

Ước chung lớn nhất và tổ hợp tuyến tính



Ví dụ 13

$$\text{ExtEuclid}(252, 198) = (18, 4, -5)$$

Gọi $\text{ExtEuclid}(\cdot, \cdot)$	a	b	d	s	t
1	252	198	18	4	-5
2	198	54	18	-1	4
3	54	36	18	1	-1
4	36	18	18	0	1
5	18	0	18	1	0

Bài tập 18

Biểu diễn ước chung lớn nhất của các cặp số sau dưới dạng tổ hợp tuyến tính của chúng

(a) 10, 11

(b) 21, 44

(c) 36, 48

(d) 34, 55

(e) 117, 213

(f) 1023, 36

Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

52

75

Số nguyên tố và Ước chung lớn nhất

Ước chung lớn nhất và tổ hợp tuyến tính



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Định lý 13

Cho các số nguyên dương a, b, c thỏa mãn $\gcd(a, b) = 1$ và $a \mid bc$. Ta có $a \mid c$

Chứng minh.

- Theo Định lý Bézout, tồn tại các số nguyên s, t thỏa mãn $\gcd(a, b) = 1 = sa + tb$
- Do $a \mid bc$, ta cũng có $a \mid tbc$
- Mặt khác, $a \mid sac$
- Suy ra, $a \mid (tb + sa)c$, hay $a \mid c$

53

75

Số nguyên tố và Ước chung lớn nhất

Ước chung lớn nhất và tổ hợp tuyến tính



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Định lý 14

Cho số nguyên dương m và các số nguyên a, b, c . Nếu $ac \equiv bc \pmod{m}$ và $\gcd(c, m) = 1$, thì $a \equiv b \pmod{m}$

Chứng minh.

- Theo định nghĩa, do $ac \equiv bc \pmod{m}$, ta có $m \mid (a - b)c$
- Kết hợp với $\gcd(c, m) = 1$ và Định lý 13, ta có $m \mid (a - b)$, nghĩa là $a \equiv b \pmod{m}$

54

75

Phương trình đồng dư

Giới thiệu



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

55

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

- Một *phương trình đồng dư (congruence)* có dạng

$$ax \equiv b \pmod{m}$$

với $a, b \in \mathbb{Z}$, $m \in \mathbb{Z}^+$, và x là một biến, được gọi là một *phương trình đồng dư tuyến tính (linear congruence)*

- Việc *giải* phương trình đồng dư nghĩa là tìm giá trị của x thỏa mãn phương trình đó
- Một *ngịch đảo (inverse)* của a theo môđun m là bất kỳ số nguyên s nào thỏa mãn $sa \equiv 1 \pmod{m}$
 - Ví dụ, 5 là một nghịch đảo của 3 theo môđun 7, vì $5 \cdot 3 \equiv 1 \pmod{7}$
 - Đôi khi ta cũng dùng ký hiệu $\bar{a}$ hoặc a^{-1} để chỉ một nghịch đảo của a
 - Chú ý rằng nếu ta có thể tìm được s thỏa mãn điều kiện trên, ta có thể giải $ax \equiv b \pmod{m}$ bằng cách nhân cả hai vế với s , nghĩa là, $sax \equiv sb \pmod{m}$, suy ra $x \equiv sb \pmod{m}$

75

Phương trình đồng dư

Giới thiệu



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Định lý 15

Nếu $\gcd(a, m) = 1$ và $m > 1$ thì tồn tại nghịch đảo s của a .
Thêm vào đó, nghịch đảo này là duy nhất theo môđun m

Chứng minh.

- Tồn tại số nguyên s thỏa mãn $sa \equiv 1 \pmod{m}$
 - Theo định lý Bézout, tồn tại các số nguyên s, t thỏa mãn $sa + tm = 1$. Do đó $sa + tm \equiv 1 \pmod{m}$
 - Do $tm \equiv 0 \pmod{m}$, ta có $sa \equiv 1 \pmod{m}$, và do đó s là một nghịch đảo của a theo môđun m
- Nếu tồn tại hai số nguyên s, r thỏa mãn $sa \equiv 1 \pmod{m}$ và $ra \equiv 1 \pmod{m}$ thì $s \equiv r \pmod{m}$
 - **Nhắc lại:** Với các số nguyên a, b, c và số nguyên dương m , nếu $ac \equiv bc \pmod{m}$ và $\gcd(c, m) = 1$ thì $a \equiv b \pmod{m}$

Bài tập 19

Chứng minh rằng nếu $\gcd(a, m) > 1$ với $a \in \mathbb{Z}$ bất kỳ và $m > 2$ thì không tồn tại một nghịch đảo của a theo môđun m

56

75

Phương trình đồng dư

Giới thiệu



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

57

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Định lý 15 cho ta một phương pháp tìm một nghịch đảo của $a \in \mathbb{Z}$ theo môđun $m \in \mathbb{Z}^+$ khi $\gcd(a, m) = 1$ và $m > 1$

Ví dụ 14

Tìm một nghịch đảo của 3 theo môđun 7

(1) Tìm các số nguyên s, t thỏa mãn $1 = s \cdot 3 + t \cdot 7$

- Thuật toán Euclid tìm ước chung lớn nhất của 3 và 7 bằng cách sử dụng phương trình

$$7 = 2 \cdot 3 + 1$$

- Từ phương trình trên, ta có

$$1 = -2 \cdot 3 + 1 \cdot 7$$

nghĩa là $s = -2$ và $t = 1$

(2) Theo Định lý 15, $s = -2$ là một nghịch đảo của 3 theo môđun 7. Chú ý rằng mọi số nguyên t thỏa mãn $t \equiv -2 \pmod{7}$ (ví dụ như 5, -9, 12, ...) đều là nghịch đảo của 3 theo môđun 7

Phương trình đồng dư

Giới thiệu



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

58

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Ví dụ 15

Giải phương trình $3x \equiv 4 \pmod{7}$

- Từ ví dụ trước, ta biết rằng -2 là một nghịch đảo của 3 theo môđun 7. Nhân cả hai vế của phương trình với -2 , ta có

$$-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}$$

- Do $-6 \equiv 1 \pmod{7}$ và $-8 \equiv 6 \pmod{7}$, nếu x là nghiệm của phương trình thì $x \equiv 6 \pmod{7}$

- Thật vậy, với mọi x thỏa mãn $x \equiv 6 \pmod{7}$

$$3x \equiv 3 \cdot 6 = 18 \equiv 4 \pmod{7}$$

Phương trình đồng dư

Giới thiệu



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

59

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Bài tập 20

Tìm nghịch đảo của a theo môđun m với

(1) $a = 4, m = 9$

(2) $a = 19, m = 141$

(3) $a = 55, m = 89$

(4) $a = 89, m = 232$

Bài tập 21

Giải các phương trình đồng dư

(1) $4x \equiv 5 \pmod{9}$

(2) $19x \equiv 4 \pmod{141}$

(3) $55x \equiv 34 \pmod{89}$

(4) $89x \equiv 2 \pmod{232}$

Phương trình đồng dư

Giới thiệu



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

60

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Bài tập 22

Cho các số nguyên dương $m_1, m_2, \dots, m_n$ thỏa mãn $m_i \geq 2$ và $\gcd(m_i, m_j) = 1$ với mọi $i \neq j$ và $1 \leq i, j \leq n$. Chứng minh rằng nếu $a \equiv b \pmod{m_i}$ với mọi $1 \leq i \leq n$, thì $a \equiv b \pmod{m}$ với $m = m_1 m_2 \dots m_n$. (**Gợi ý:** Chứng minh với $n = 2$)

75

Phương trình đồng dư

Định lý phần dư Trung Hoa



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

61

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Định lý phần dư Trung Hoa (The Chinese Remainder Theorem) nói rằng nếu các môđun của một hệ các phương trình đồng dư tuyến tính là đôi một nguyên tố cùng nhau thì hệ phương trình có nghiệm duy nhất theo môđun tích của các môđun của từng phương trình

Định lý 16: Định lý phần dư Trung Hoa

Cho các số nguyên dương $m_1, m_2, \dots, m_n$ thỏa mãn $m_i \geq 2$ và $\gcd(m_i, m_j) = 1$ với mọi $i \neq j$ và $1 \leq i, j \leq n$. Cho các số nguyên bất kỳ $a_1, a_2, \dots, a_n$. Hệ phương trình

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

$$\vdots$$

$$x \equiv a_n \pmod{m_n}$$

có nghiệm duy nhất theo môđun $m = m_1 m_2 \dots m_n$. (Nghĩa là, tồn tại một nghiệm x với $0 \leq x < m$, và tất cả các nghiệm khác đồng dư với x theo môđun m)

Phương trình đồng dư

Định lý phần dư Trung Hoa



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

62

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Chứng minh (tồn tại).

- Đặt $M_i = m/m_i$ ($1 \leq i \leq n$). Do đó $\gcd(M_i, m_i) = 1$
- Theo Định lý 15, tồn tại số nguyên y_i sao cho $y_i M_i \equiv 1 \pmod{m_i}$
- Đặt $x = \sum_{i=1}^n a_i y_i M_i = a_1 y_1 M_1 + a_2 y_2 M_2 + \cdots + a_n y_n M_n$
- Do $m_i \mid M_k$ với mọi $k \neq i$, $M_k \equiv 0 \pmod{m_i}$, do đó $x \equiv a_i y_i M_i \equiv a_i \pmod{m_i}$ với mọi i . Do đó x là nghiệm của hệ phương trình đã cho



Bài tập 23

Hoàn thành Chứng minh của Định lý phần dư Trung Hoa bằng cách chỉ ra nghiệm x của hệ phương trình đã cho là duy nhất theo môđun $m = m_1 m_2 \dots m_n$ (**Gợi ý:** Giả sử x và y là hai nghiệm phân biệt của hệ phương trình đã cho. Chứng minh rằng $m_i \mid (x - y)$ với mọi $1 \leq i \leq n$. Sử dụng Bài tập 22 để kết luận rằng $m \mid (x - y)$)

Phương trình đồng dư

Định lý phần dư Trung Hoa



Ví dụ 16 (Sử dụng Chứng minh của Định lý Phần dư Trung Hoa)

Giải hệ phương trình

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

$$x \equiv 5 \pmod{7}$$

Chú ý: 3, 5, 7 là dãy các số nguyên ≥ 2 và đôi một nguyên tố cùng nhau

- $m = m_1 m_2 m_3 = 3 \cdot 5 \cdot 7 = 105$
- $M_1 = m/m_1 = 35$ và $y_1 = 2$ là một nghịch đảo của M_1 theo môđun $m_1 = 3$
- $M_2 = m/m_2 = 21$ và $y_2 = 1$ là một nghịch đảo của M_2 theo môđun $m_2 = 5$
- $M_3 = m/m_3 = 15$ và $y_3 = 1$ là một nghịch đảo của M_3 theo môđun $m_3 = 7$
- $x = \sum_{i=1}^3 a_i y_i M_i = 2 \cdot 2 \cdot 35 + 3 \cdot 1 \cdot 21 + 5 \cdot 1 \cdot 15 = 278 \equiv 68 \pmod{105}$

Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

63

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

75

Phương trình đồng dư

Định lý phần dư Trung Hoa



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Ví dụ 17 (Phương pháp thay ngược)

Giải hệ phương trình

$$x \equiv 2 \pmod{3} \quad (1)$$

$$x \equiv 3 \pmod{5} \quad (2)$$

$$x \equiv 5 \pmod{7} \quad (3)$$

- Từ (1), tồn tại $t \in \mathbb{Z}$ sao cho $x = 3t + 2$
- Thay vào (2), ta có $3t + 2 \equiv 3 \pmod{5}$, suy ra $3t \equiv 1 \pmod{5}$, do đó $t \equiv 2 \pmod{5}$. Do đó, tồn tại $u \in \mathbb{Z}$ sao cho $t = 5u + 2$. Suy ra, $x = 3t + 2 = 3(5u + 2) + 2 = 15u + 8$
- Thay vào (3), ta có $15u + 8 \equiv 5 \pmod{7}$, suy ra $15u \equiv -3 \pmod{7}$, do đó $u \equiv 4 \pmod{7}$. Do đó, tồn tại $v \in \mathbb{Z}$ sao cho $u = 7v + 4$
- Suy ra $x = 15u + 8 = 15(7v + 4) + 8 = 105v + 68$. Do đó, $x \equiv 68 \pmod{105}$

Phương trình đồng dư

Định lý phần dư Trung Hoa



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

65

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Bài tập 24

Giải hệ phương trình sau bằng các phương pháp đã đề cập

$$x \equiv 1 \pmod{5} \quad (4)$$

$$x \equiv 2 \pmod{6} \quad (5)$$

$$x \equiv 3 \pmod{7} \quad (6)$$

Bài tập 25

Giải hệ phương trình sau bằng các phương pháp đã đề cập

$$x \equiv 2 \pmod{3} \quad (7)$$

$$x \equiv 1 \pmod{4} \quad (8)$$

$$x \equiv 3 \pmod{5} \quad (9)$$

Phương trình đồng dư

Định lý phần dư Trung Hoa



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Bài tập 26

Giải hệ phương trình sau bằng các phương pháp đã đề cập

$$x \equiv 1 \pmod{2} \quad (10)$$

$$x \equiv 2 \pmod{3} \quad (11)$$

$$x \equiv 3 \pmod{5} \quad (12)$$

$$x \equiv 4 \pmod{11} \quad (13)$$

Bài tập 27

Những số nguyên nào chia 2 dư 1 và chia 3 cũng dư 1?

66

75

Phương trình đồng dư

Định lý phần dư Trung Hoa



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Bài tập 28 (*)

Giải hệ phương trình

$$x \equiv 5 \pmod{6} \quad (14)$$

$$x \equiv 3 \pmod{10} \quad (15)$$

$$x \equiv 8 \pmod{15} \quad (16)$$

Chú ý: 6, 10, và 15 *không* đôi một nguyên tố cùng nhau

Bài tập 29 (*)

Giải hệ phương trình

$$y \equiv 5x - 3 \pmod{7} \quad (17)$$

$$y \equiv 3x + 2 \pmod{7} \quad (18)$$

67

75

Phương trình đồng dư

Định lý phần dư Trung Hoa



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Định lý phần dư Trung Hoa cho ta một cách thực hiện các tính toán số học với các số nguyên lớn

- Theo Định lý, một số nguyên a với $0 \leq a < m = m_1 m_2 \dots m_n$ trong đó $\gcd(m_i, m_j) = 1$ với mọi $i \neq j$, $1 \leq i, j \leq n$, có thể được biểu diễn thông qua bộ $(a \bmod m_1, a \bmod m_2, \dots, a \bmod m_n)$
- Để thực hiện tính toán với các số nguyên lớn được biểu diễn theo cách này
 - Thực hiện tính toán riêng biệt cho từng bộ
 - Mỗi tính toán có thể được thực hiện trong cùng một máy tính hoặc thực hiện song song
 - Xuất kết quả đầu ra bằng cách giải hệ phương trình đồng dư
 - Có thể thực hiện khi m luôn lớn hơn kết quả đầu ra mong muốn

68

75

Phương trình đồng dư

Định lý Fermat nhỏ



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Định lý 17: Định lý Fermat nhỏ

Nếu p là một số nguyên tố và a là một số nguyên không chia hết cho p , thì $a^{p-1} \equiv 1 \pmod{p}$. Thêm vào đó, với mọi số nguyên a , ta có $a^p \equiv a \pmod{p}$

Bài tập 30 (Chứng minh Định lý Fermat nhỏ)

Nhắc lại: Với các số nguyên a, b, c và số nguyên dương m , nếu $ac \equiv bc \pmod{m}$ và $\gcd(c, m) = 1$ thì $a \equiv b \pmod{m}$.

- Giả sử a không chia hết cho p . Chứng minh rằng không có hai số nguyên nào trong số các số $1 \cdot a, 2 \cdot a, \dots, (p-1) \cdot a$ là đồng dư theo môđun p
- Từ phần (a), kết luận rằng tích các số $1, 2, \dots, p-1$ đồng dư với tích các số $a, 2a, \dots, (p-1)a$ theo môđun p . Sử dụng điều này để chứng minh rằng $(p-1)! \equiv a^{p-1}(p-1)! \pmod{p}$
- Chỉ ra từ phần (b) rằng $a^{p-1} \equiv 1 \pmod{p}$ nếu a không chia hết cho p . (**Gợi ý:** Xem lại phần chứng minh Định lý cơ bản của số học. Chứng minh $p \nmid (p-1)!$ và áp dụng mệnh đề trên)

69

75

Phương trình đồng dư

Định lý Fermat nhỏ



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Ví dụ 18 (Tìm số dư của phép chia cho số nguyên tố)

Tìm $7^{222} \bmod 11$

- Theo Định lý Fermat nhỏ, ta có $7^{10} \equiv 1 \pmod{11}$
- Do đó, $(7^{10})^k \equiv 1 \pmod{11}$ với mọi $k \in \mathbb{Z}$
- Mặt khác, $7^{222} = 7^{10 \cdot 22 + 2} = (7^{10})^{22} \cdot 7^2 \equiv 49 \equiv 5 \pmod{11}$

Bài tập 31

Sử dụng Định lý Fermat nhỏ để tính

- (a) $7^{121} \bmod 13$
- (b) $23^{1002} \bmod 41$
- (c) nghịch đảo của 5^{39} theo môđun 41

70

75

Phương trình đồng dư

Định lý Fermat nhỏ



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

Thuật toán mã hóa RSA

Bài tập 32

- (a) Sử dụng Định lý Fermat nhỏ để tính $5^{2003} \bmod 7$, $5^{2003} \bmod 11$, và $5^{2003} \bmod 13$
- (b) Sử dụng kết quả từ phần (a) và Định lý phần dư Trung Hoa để tính $5^{2003} \bmod 1001$ (Chú ý rằng $1001 = 7 \cdot 11 \cdot 13$)

Bài tập 33

Sử dụng sự trợ giúp từ Định lý Fermat nhỏ, hãy chứng minh rằng 42 là ước của $n^7 - n$

71

75

Thuật toán mã hóa RSA

Mật mã khóa công khai



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán modulo

Định nghĩa và tính chất cơ bản

Đồng dư theo modulo

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa modulo

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

72

Thuật toán mã hóa RSA

- Trong **mật mã khóa bí mật (private key cryptography)**, cùng một khóa bí mật được sử dụng cho cả quá trình mã hóa và giải mã
 - Thách thức lớn nhất là **làm thế nào để trao đổi khóa bí mật một cách an toàn**
- Trong **mật mã khóa công khai (public key cryptography)**, hai khóa riêng biệt được sử dụng: một khóa để mã hóa và một khóa khác để giải mã
 - Bất kỳ ai có khóa công khai đều có thể mã hóa thông điệp, nhưng chỉ người sở hữu khóa bí mật mới có thể giải mã
 - Khi người sở hữu khóa bí mật mã hóa thông tin bằng khóa bí mật của mình, bất kỳ ai có khóa công khai đều có thể giải mã và xác minh nguồn gốc thông điệp (Đây chính là nguyên lý của chữ ký điện tử)
- RSA là hệ thống mã hóa khóa công khai được biết đến và sử dụng rộng rãi nhất

Thuật toán mã hóa RSA

RSA - Rivest-Shamir-Adleman



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

73

Thuật toán mã hóa RSA

75

- Chọn hai số nguyên tố lớn phân biệt p, q
- Đặt $n = pq$ và $k = (p - 1)(q - 1)$
- Chọn số nguyên e thỏa mãn $1 < e < k$ và $\gcd(e, k) = 1$
- Tính nghịch đảo d của e theo môđun k , nghĩa là $de \equiv 1 \pmod{k}$
- **Khóa công khai:** (n, e)
- **Khóa bí mật:** (n, d)
- **Mã hóa:**
 - Chuyển thông điệp M cần mã hóa thành số nguyên m , $0 \leq m < n$
 - Thông điệp mã hóa c được tính bằng $c = m^e \pmod{n}$ (Việc này có thể được thực hiện một cách hiệu quả. Xem bài giảng trước)
- **Giải mã:**
 - Tính $m = c^d \pmod{n}$
 - Chuyển m từ số nguyên sang thông điệp M ban đầu

Thuật toán mã hóa RSA

RSA - Rivest-Shamir-Adleman



Ví dụ 19

- $n = pq = 43 \cdot 59 = 2537$, $k = 42 \cdot 58 = 2436$
- Chọn $e = 13$: $1 < e < k$ và $\gcd(13, 2436) = 1$
- $d = 937$ là nghịch đảo của 13 theo môđun 2436
- **Khóa công khai:** $(2537, 13)$
- **Khóa bí mật:** $(2537, 937)$

Mã hóa và Giải mã

- Chuyển thông điệp $M = \text{STOP}$ gồm các chữ cái thành số nguyên bằng cách gán mỗi chữ cái bằng thứ tự trong bảng chữ cái tiếng Anh trừ đi 1: $\text{ST} \Rightarrow 1819$ và $\text{OP} \Rightarrow 1415$
- $1819^{13} \bmod 2537 = 2081$ và $1415^{13} \bmod 2537 = 2182$
- Thông điệp mã hóa là 2081 2182
- Ví dụ nếu nhận được thông điệp 0981 0461
- $0981^{937} \bmod 2537 = 0704$ và $0461^{937} \bmod 2537 = 1115$
- Thông điệp giải mã là HELP

Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

74

Thuật toán mã hóa RSA

75

Thuật toán mã hóa RSA

RSA - Rivest-Shamir-Adleman



Tính đúng đắn của quá trình giải mã.

Ta chứng minh nếu $c = m^e \bmod n$ thì $m = c^d \bmod n$.

- Ta có $c^d = (m^e)^d \equiv m^{ed} \pmod{n}$
- Theo cách xây dựng, $ed \equiv 1 \pmod{k}$ với $k = (p-1)(q-1)$. Do đó tồn tại số nguyên h thỏa mãn $ed - 1 = h(p-1)(q-1)$
- Ta xét $m^{ed} \bmod p$. Nếu $p \nmid m$ thì theo Định lý Fermat nhỏ, ta có

$$\begin{aligned} m^{ed} &= m^{h(p-1)(q-1)} m = (m^{p-1})^{h(q-1)} m \\ &\equiv 1^{h(q-1)} m \equiv m \pmod{p} \end{aligned}$$

Nếu $p \mid m$, ta có $m^{ed} \equiv 0 \equiv m \pmod{p}$. Tóm lại, $m^{ed} \equiv m \pmod{p}$. Tương tự, ta có $m^{ed} \equiv m \pmod{q}$

- Do $\gcd(p, q) = 1$, sử dụng Định lý phần dư Trung Hoa, ta có $m^{ed} \equiv m \pmod{pq}$
 - Do $\gcd(p, q) = 1$, theo Định lý Bézout, tồn tại $s, t \in \mathbb{Z}$ thỏa mãn $sp + tq = 1$. Đặt $x = m \cdot sp + m \cdot tq$ thì $x \bmod p = (m \cdot sp + m \cdot (1 - sp)) \bmod p = m \bmod p$. Suy ra $x \equiv m \pmod{p}$. Tương tự, $x \equiv m \pmod{q}$
 - Theo Định lý phần dư Trung Hoa, $x \equiv m^{ed} \pmod{pq}$, hay $m^{ed} \equiv m \pmod{pq} \equiv m \pmod{n}$



Lý thuyết số cơ bản

Hoàng Anh Đức

Giới thiệu

Tính chia hết và phép toán môđun

Định nghĩa và tính chất cơ bản

Đồng dư theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b -phân

Cộng và nhân các số nhị phân

Biểu diễn các số nguyên âm theo hệ nhị phân

Tính lũy thừa môđun

Số nguyên tố và Ước chung lớn nhất

Số nguyên tố

Ước chung lớn nhất

Phương trình đồng dư

Giới thiệu

Định lý phần dư Trung Hoa

Định lý Fermat nhỏ

75

Thuật toán mã hóa RSA

75

Part I

Phụ lục

Nội dung



Lý thuyết số cơ bản

Hoàng Anh Đức

Một số lỗi thường gặp

Tài liệu tham khảo

Một số lỗi thường gặp

Một số lỗi thường gặp



Lý thuyết số cơ bản

Hoàng Anh Đức

2 Một số lỗi thường gặp

Tài liệu tham khảo

Chú ý

Tham khảo từ tài liệu “Common Mistakes in Discrete Mathematics” (https://highereducation.com/sites/default/files/125967651x/1106131/Common_Mistakes_in_Discrete_Math.pdf)

(a) Sai lầm khi làm cho $a \bmod m$ trở thành số âm

- Ví dụ, $-16 \bmod 5$ là 4, không phải -1
- Tương tự, phép chia lấy nguyên $a \div m$ luôn được tính bằng cách làm tròn xuống. Ví dụ, $-16 \div 5$ là -4 , không phải -3

(b) Nhầm lẫn giữa a/b và $a \mid b$

- Dấu gạch chéo $/$ là một phép toán (phép chia), và kết quả của phép toán là một số. Ví dụ, $6/3$ là số 2

Một số lỗi thường gặp (tiếp)



Lý thuyết số cơ bản

Hoàng Anh Đức

3 Một số lỗi thường gặp

Tài liệu tham khảo

- Dấu gạch đứng $|$ là động từ của một câu. Ví dụ, $3 | 6$ là khẳng định rằng 3 là ước số của 6; nó không nói đến kết quả của việc thực hiện phép chia

(c) Nhầm lẫn giữa thứ tự trong biểu thức $a | b$

- Viết $a | b$ khi thực sự muốn viết $b | a$. Đúng là $3 | 6$, nhưng không đúng là $6 | 3$

(d) Lỗi xem số 1 là số nguyên tố

- Đây là vấn đề quy ước, nhưng theo định nghĩa, số 1 không phải là số nguyên tố và cũng không phải là hợp số

(e) Quên rằng mọi số nguyên dương đều là ước của 0

- Do đó $\gcd(a, 0) = a$ với mọi số nguyên dương a . Ví dụ, $\gcd(6, 0) = 6$
- Tất nhiên, 0 không là ước của bất kỳ số khác 0 nào, và phép chia cho 0 là không xác định

(f) Khi thực hiện thuật toán Euclid, sử dụng thương cuối cùng là kết quả

Một số lỗi thường gặp (tiếp)



Lý thuyết số cơ bản

Hoàng Anh Đức

4

Một số lỗi thường gặp

Tài liệu tham khảo

- Đúng ra phải sử dụng số chia cuối cùng làm ước chung lớn nhất, không phải thương cuối cùng.
- Ví dụ, nếu bước cuối là chia 8 cho 4, cho thương là 2 và số dư 0, thì ước chung lớn nhất là số chia cuối cùng (tức là 4) chứ không phải là thương cuối cùng (2)

(g) Giả định sai rằng mọi quy tắc đúng với đẳng thức cũng đúng với đồng dư

- Ví dụ, không đúng rằng nếu $r \equiv s \pmod{m}$, thì $a^r \equiv a^s \pmod{m}$.
- Hãy thử với $m = 3$, $a = 2$, $r = 1$, và $s = 4$.
- Cũng lưu ý rằng dù $8 \equiv 14 \pmod{6}$, nhưng sẽ sai khi chia cả hai vế cho 2 và khẳng định $4 \equiv 7 \pmod{6}$

5



Agrawal, Manindra, Neeraj Kayal, and Nitin Saxena (2004). “PRIMES is in P”. In: *Annals of Mathematics* 160.2, pp. 781–793. DOI: 10.4007/annals.2004.160.781.

1. Logic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Đệ quy
4. Thuật toán I: Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp
5. Thuật toán II: Thuật toán tham lam, thuật toán đệ quy
6. Lý thuyết số cơ bản
7. Các phương pháp đếm
8. Lý thuyết đồ thị I: Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông
9. Lý thuyết đồ thị II: Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị
10. Lý thuyết đồ thị III: Cây
11. Đại số Boole

VNU-HUS MAT3500: Toán rời rạc

Các phương pháp đếm

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn



Nội dung



Các nguyên lý đếm cơ bản

Quy tắc song ánh
Quy tắc nhân và Quy tắc cộng
Nguyên lý bù trừ
Quy tắc chia

Nguyên lý chuồng bồ câu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu
Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal
Định lý nhị thức
Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh
Quy tắc nhân và Quy tắc cộng
Nguyên lý bù trừ
Quy tắc chia

Nguyên lý chuồng bồ câu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu
Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal
Định lý nhị thức
Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Quy tắc song ánh



Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

2

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuồng bồ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

- **Quy tắc song ánh (The Bijection Rule):** Nếu $f : A \rightarrow B$ là một song ánh, trong đó A và B là các tập hữu hạn, thì $|A| = |B|$

Ví dụ 1

- Cho tập hợp $U = \{u_1, u_2, \dots, u_n\}$ gồm n phần tử và tập hợp Σ_n^* các chuỗi nhị phân độ dài n
- $f : \mathcal{P}(U) \rightarrow \Sigma_n^*$ được định nghĩa như sau: Với mỗi $A \in \mathcal{P}(U)$, $f(A) = x_1 x_2 \dots x_n$ trong đó $x_i = 1$ nếu $u_i \in A$ và $x_i = 0$ nếu $u_i \notin A$
- Do f là song ánh (**Tại sao?**), $|\mathcal{P}(U)| = |\Sigma_n^*|$ (số tập con của U bằng với số chuỗi nhị phân độ dài n)

Các nguyên lý đếm cơ bản

Quy tắc nhân và Quy tắc cộng



■ Quy tắc nhân (The Product Rule):

- Giả sử một công việc được chia nhỏ ra thành n giai đoạn liên tiếp nhau
 - Giai đoạn thứ 1 có m_1 cách thực hiện
 - Với mỗi cách thực hiện giai đoạn thứ 1, có m_2 cách thực hiện giai đoạn thứ 2
 - Với mỗi cách thực hiện các giai đoạn thứ 1 và 2, có m_3 cách thực hiện giai đoạn thứ 3
 - ...
 - Với mỗi cách thực hiện các giai đoạn thứ 1, 2, $\dots$, $n - 1$, có m_n cách thực hiện giai đoạn thứ n
- Có $m_1 m_2 \dots m_n$ cách thực hiện công việc

■ Quy tắc cộng (The Sum Rule):

- Có n biện pháp khác nhau để thực hiện một công việc
- Cách thực hiện biện pháp thứ i luôn luôn khác cách thực hiện biện pháp thứ j với mọi $i \neq j$ và $1 \leq i, j \leq n$
- Nếu biện pháp thứ i có m_i cách thực hiện ($1 \leq i \leq n$) thì ta có $m_1 + m_2 + \dots + m_n$ cách thực hiện công việc

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Quy tắc nhân và Quy tắc cộng



Các quy tắc đếm cơ bản có thể được biểu diễn theo ngôn ngữ tập hợp

- **Quy tắc nhân (The Product Rule):** Cho các tập hữu hạn $A_1, A_2, \dots, A_n$ trong đó $|A_i| = m_i$ với $1 \leq i \leq n$. Ta có

$$|A_1 \times A_2 \times \dots \times A_n| = |A_1| \cdot |A_2| \cdot \dots \cdot |A_n| = m_1 m_2 \dots m_n$$

- **Quy tắc cộng (The Sum Rule):** Cho các tập hữu hạn đôi một rời nhau $A_1, A_2, \dots, A_n$, ($A_i \cap A_j = \emptyset$ với mọi $i \neq j$, $1 \leq i, j \leq n$) và $|A_i| = m_i$ với $1 \leq i \leq n$. Ta có

$$|A_1 \cup A_2 \cup \dots \cup A_n| = |A_1| + |A_2| + \dots + |A_n| = m_1 + m_2 + \dots + m_n$$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Quy tắc nhân và Quy tắc cộng



Ví dụ 2 (Quy tắc nhân)

Có bao nhiêu chuỗi nhị phân độ dài 7?

- Giả sử chuỗi $x = x_1x_2 \dots x_7$ là một chuỗi nhị phân độ dài 7
- Để xây dựng x , ta lần lượt chọn giá trị cho $x_1, x_2, \dots, x_7$
 - có 2 cách chọn x_1 (0 hoặc 1)
 - với mỗi giá trị của x_1 , có 2 cách chọn x_2 (0 hoặc 1)
 - ...
 - với mỗi giá trị của $x_1, \dots, x_6$, có 2 cách chọn x_7 (0 hoặc 1)
- Do đó có 2^7 chuỗi nhị phân độ dài 7

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Quy tắc nhân và Quy tắc cộng



Ví dụ 3 (Quy tắc nhân)

Tập hợp n phần tử $S = \{x_1, x_2, \dots, x_n\}$ có bao nhiêu tập con?

- Một tập con của S có thể được xây dựng thông qua n bước liên tiếp
 - chọn x_1 hoặc không chọn
 - chọn x_2 hoặc không chọn
 - ...
 - chọn x_n hoặc không chọn
- Mỗi bước có thể được thực hiện bằng 2 cách
- Do đó có 2^n tập con của S

Bài tập 1

Có bao nhiêu hàm $f: A \rightarrow B$ với A và B lần lượt là các tập hữu hạn gồm m và n phần tử?

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Quy tắc nhân và Quy tắc cộng



Ví dụ 4 (Quy tắc cộng)

Một sinh viên có thể chọn một bài thực hành máy tính từ một trong ba danh sách tương ứng có 23, 15, và 19 bài. Giả thiết rằng không có hai bài nào giống nhau. Có bao nhiêu cách chọn bài thực hành?

- Có 23 cách chọn bài thực hành từ danh sách thứ nhất
- Có 15 cách chọn bài thực hành từ danh sách thứ hai
- Có 19 cách chọn bài thực hành từ danh sách thứ ba
- Do không có hai bài nào giống nhau, số cách chọn bài thực hành là $23 + 15 + 19 = 57$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Quy tắc nhân và Quy tắc cộng



Ví dụ 5 (Quy tắc cộng)

Có bao nhiêu chuỗi nhị phân độ dài 7 có chính xác hai số 1?

- Trong một chuỗi nhị phân $x_1x_2 \dots x_7$ độ dài 7 có chính xác hai số 1

- Ở các vị trí i và j nào đó với $1 \leq i < j \leq 7$, $x_i = x_j = 1$
- Ở các vị trí $k \notin \{i, j\}$, $x_k = 0$

x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	0	1	0	1	0	0
		i		j		

- Ứng với mỗi vị trí i của số 1 đầu tiên, có $7 - i$ vị trí j có thể cho số 1 thứ hai

- Với $i = 1$, có $7 - 1 = 6$ lựa chọn cho j

...

- Với $i = 7$, có $7 - 7 = 0$ lựa chọn cho j

- Theo quy tắc cộng, số các chuỗi nhị phân độ dài 7 có chính

$$\text{xác hai số 1 là } \sum_{i=1}^7 (7 - i) = \sum_{i=0}^6 i = 6(6 + 1)/2 = 21$$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Quy tắc nhân và Quy tắc cộng



Ví dụ 6

Giả sử tên các biến trong một ngôn ngữ lập trình chỉ có thể là một chữ cái viết hoa hoặc một chữ cái viết hoa theo sau bởi một chữ số. Giả sử ta sử dụng bảng chữ cái tiếng Anh và các chữ số trong hệ thập phân. Có tất cả bao nhiêu tên biến trong ngôn ngữ lập trình này?

- Một tên biến có dạng x hoặc $x\alpha$ với $x \in \{A, B, \dots, Z\}$ và $\alpha \in \{0, 1, \dots, 9\}$
- Nếu tên biến có dạng x , có 26 cách chọn giá trị của x
- Nếu tên biến có dạng $x\alpha$, có 26 cách chọn giá trị của x , và ứng với mỗi giá trị của x có 10 cách chọn giá trị của α . Theo quy tắc nhân, có $26 \times 10 = 260$ tên biến có dạng $x\alpha$
- Theo quy tắc cộng

$$\begin{aligned}\text{số tên biến} &= \text{số tên biến dạng } x + \text{số tên biến dạng } x\alpha \\ &= 26 + 260 = 286\end{aligned}$$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Quy tắc nhân và Quy tắc cộng



Ví dụ 7

Các *dịch vụ rút gọn đường dẫn (URL-shortening service)* như bit.ly hay tinyurl.com cho phép người dùng thu gọn một đường dẫn dài thành một dãy các ký tự ngắn hơn rất nhiều. Ví dụ đường dẫn tới bài giảng trên trang web môn học <https://hoanganhduc.github.io/teaching/VNU-HUS/2025/winter/MAT3500-3/Counting.pdf> sau khi rút gọn thông qua bit.ly là <https://bit.ly/4oTFUGF>.

Giả sử các đường dẫn sau khi rút gọn gồm có [https://bit.ly/](https://bit.ly) kèm theo một chuỗi 7 ký tự, mỗi ký tự chỉ có thể là một chữ số thập phân, một chữ cái viết hoa, hoặc một chữ cái viết thường trong bảng chữ cái tiếng Anh. Dịch vụ bit.ly có thể cung cấp tối đa bao nhiêu đường dẫn rút gọn?

- Mỗi đường dẫn tương ứng với một chuỗi ký tự $x_1x_2 \dots x_7$ trong đó $x_i \in C = \{0, \dots, 9\} \cup \{A, \dots, Z\} \cup \{a, \dots, z\}$, $1 \leq i \leq 7$ (Quy tắc song ánh)
 - Có $|C| = 10 + 26 + 26 = 62$ cách chọn giá trị cho x_1 (Quy tắc cộng)
 - Với mỗi giá trị của x_1 , có $|C| = 62$ cách chọn giá trị cho x_2
 - ...
 - Với mỗi giá trị của $x_1, \dots, x_6$, có $|C| = 62$ cách chọn giá trị cho x_7
- Theo quy tắc nhân, có $|C|^7 = 62^7 = 3\,521\,614\,606\,208$ đường dẫn rút gọn

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chồng bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

10

71

Các nguyên lý đếm cơ bản

Quy tắc nhân và Quy tắc cộng



Bài tập 2

Địa chỉ kiểm soát truy cập phương tiện truyền thông (MAC (media access control) address), hay còn gọi là *địa chỉ MAC*, là một định danh duy nhất được gán cho một card mạng (network adapter), ví dụ như card mạng có dây (ethernet card) hoặc card mạng không dây (wireless card). Địa chỉ này gồm một dãy sáu cặp các chữ số thập lục phân (nghĩa là các chữ số 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). Ví dụ, F7:DE:A1:B6:C4:33 là một địa chỉ MAC. (Các cặp số thường được phân tách bởi dấu hai chấm.) Có tất cả bao nhiêu địa chỉ MAC?

Bài tập 3

Theo quy định của Bộ Thông Tin & Truyền Thông Việt Nam, tất cả các số thuê bao điện thoại di động cần có 10 chữ số, trong đó ba chữ số đầu tiên đại biểu cho nhà mạng cung cấp dịch vụ. Ví dụ, nhà mạng Viettel hiện tại sở hữu 12 đầu số di động: 086, 096 – 098, và 032 – 039. Viettel có thể cung cấp tối đa bao nhiêu số điện thoại di động?

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Quy tắc nhân và Quy tắc cộng



Bài tập 4

Giả sử một mật khẩu cho một hệ thống máy tính phải có ít nhất 8, nhưng không quá 12 ký tự, trong đó mỗi ký tự trong mật khẩu là một chữ cái tiếng Anh viết thường, một chữ cái tiếng Anh viết hoa, một chữ số, hoặc một trong sáu ký tự đặc biệt *, >, <, !, +, và =.

- (a) Có bao nhiêu mật khẩu khác nhau có sẵn cho hệ thống máy tính này?
- (b) Có bao nhiêu mật khẩu trong số này chứa ít nhất một lần xuất hiện của ít nhất một trong sáu ký tự đặc biệt?
- (c) Sử dụng câu trả lời của bạn cho phần (a), hãy xác định xem một hacker sẽ mất bao lâu để thử tất cả các mật khẩu có thể, giả sử rằng hacker mất một nano giây (nanosecond) để kiểm tra mỗi mật khẩu có thể. (Một nano giây là một phần tỷ giây, hoặc 10^{-9} giây)

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

12

71

Các nguyên lý đếm cơ bản

Quy tắc nhân và Quy tắc cộng



Bài tập 5

Tên của một biến trong ngôn ngữ lập trình C là một chuỗi có thể chứa các chữ cái viết hoa, chữ cái viết thường, chữ số, hoặc dấu gạch dưới. Hơn nữa, ký tự đầu tiên trong chuỗi phải là một chữ cái, viết hoa hoặc viết thường, hoặc dấu gạch dưới. Nếu tên của một biến được xác định bởi tám ký tự đầu tiên, có bao nhiêu biến khác nhau có thể được đặt tên trong C? (Lưu ý rằng tên của một biến có thể chứa ít hơn tám ký tự.)

Bài tập 6

Tên của một biến trong ngôn ngữ lập trình JAVA là một chuỗi có độ dài từ 1 đến 65535 ký tự (bao gồm cả hai giá trị), trong đó mỗi ký tự có thể là một chữ cái viết hoa, một chữ cái viết thường, dấu đô la (\$), dấu gạch dưới, hoặc một chữ số, ngoại trừ ký tự đầu tiên không được là một chữ số. Xác định số lượng tên biến khác nhau trong JAVA

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

13

71

Các nguyên lý đếm cơ bản

Nguyên lý bù trừ

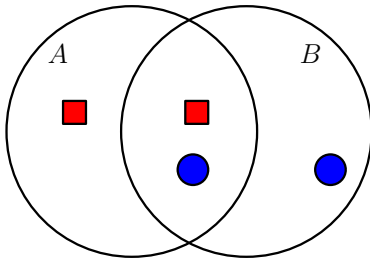


■ Nguyên lý bù trừ (Inclusion-Exclusion Principle): (hay Quy tắc trừ (The Subtraction Rule))

- Có hai biện pháp khác nhau để thực hiện một công việc
- Biện pháp thứ nhất có m cách thực hiện
- Biện pháp thứ hai có n cách thực hiện
- Có k cách thực hiện đồng thời hai biện pháp
- Số cách thực hiện công việc là $m + n - k$

■ Nguyên lý bù trừ (Inclusion-Exclusion Principle): Cho các tập hữu hạn A, B trong đó $|A| = m$ và $|B| = n$. Ta có

$$|A \cup B| = |A| + |B| - |A \cap B|$$



Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

14

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuồng bồ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Nguyên lý bù trừ



Ví dụ 8

Có bao nhiêu chuỗi nhị phân độ dài 8 bắt đầu với 1 hoặc kết thúc với 00?

- Gọi A là tập các chuỗi nhị phân độ dài 8 có dạng $1x_2 \dots x_8$ (bắt đầu với 1) và B là tập các chuỗi nhị phân độ dài 8 có dạng $x_1x_2 \dots x_600$ (kết thúc với 00)
- Tập các chuỗi nhị phân độ dài 8 bắt đầu với 1 hoặc kết thúc với 00 là $A \cup B$
- Theo nguyên lý bù trừ, $|A \cup B| = |A| + |B| - |A \cap B|$
 - $|A|$: Số chuỗi nhị phân độ dài 8 có dạng $1x_2 \dots x_8$ là 2^7
 - $|B|$: Số chuỗi nhị phân độ dài 8 có dạng $x_1x_2 \dots x_600$ là 2^6
 - $|A \cap B|$: Số chuỗi nhị phân độ dài 8 có dạng $1x_2 \dots x_600$ là 2^5

Số chuỗi nhị phân độ dài 8 bắt đầu với 1 hoặc kết thúc với 00 là $|A \cup B| = 2^7 + 2^6 - 2^5 = 160$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

15 Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chồng bỏ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Nguyên lý bù trừ



Ví dụ 9

Ở ngân hàng X , khách hàng có thể sử dụng mã **PIN** (*Personal Identification Number*) gồm 4 chữ số thập phân để truy cập tài khoản từ máy rút tiền tự động thông qua thẻ rút tiền. Ngân hàng X đặc biệt yêu cầu các mã PIN không thể bắt đầu hoặc kết thúc với ba chữ số liên tiếp giống nhau (ví dụ, dãy 7770 hoặc 0111 là không hợp lệ). Có tất cả bao nhiêu dãy PIN không hợp lệ?

- Gọi S là tập hợp tất cả các dãy PIN bắt đầu với ba chữ số giống nhau và E là tập hợp tất cả các dãy PIN kết thúc với ba chữ số giống nhau. Tập hợp các dãy PIN không hợp lệ là $S \cup E$
- Theo nguyên lý bù trừ, $|S \cup E| = |S| + |E| - |S \cap E|$
 - $|S|$: Số dãy PIN có dạng $xxxxy$ với $x, y \in \{0, \dots, 9\}$ là 10^2 (có 10 cách chọn x , và ứng với mỗi giá trị của x có 10 cách chọn y)
 - $|E|$: Số dãy PIN có dạng $xyyyy$ với $x, y \in \{0, \dots, 9\}$ là 10^2 (có 10 cách chọn x , và ứng với mỗi giá trị của x có 10 cách chọn y)
 - $|S \cap E|$: Một dãy PIN $xyztt$ thuộc $S \cap E$ khi và chỉ khi $x = y = z$ (thuộc S) và $y = z = t$ (thuộc E), nghĩa là $x = y = z = t$. Mỗi dãy thuộc $S \cap E$ do đó có dạng $xxxx$ và có 10 dãy dạng này (có 10 cách chọn x)
- Số dãy PIN không hợp lệ là $|S \cup E| = 10^2 + 10^2 - 10 = 190$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

16

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chồng bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Nguyên lý bù trừ



Ví dụ 10

Có bao nhiêu chuỗi nhị phân độ dài 10 có chứa 00000 hoặc 11111?

- Gọi A là tập các chuỗi nhị phân độ dài 10 có chứa 00000 và B là tập các chuỗi nhị phân độ dài 10 có chứa 11111. $A \cup B$ là tập chuỗi nhị phân độ dài 10 có chứa 00000 hoặc 11111
- Theo nguyên lý bù trừ, $|A \cup B| = |A| + |B| - |A \cap B|$
- Trước tiên, ta tính $|A|$. Mỗi chuỗi nhị phân độ dài 10 có chứa 00000 thuộc chính xác một trong các dạng: $00000x_6x_7x_8x_9x_{10}$, $100000x_7x_8x_9x_{10}$, $x_100000x_8x_9x_{10}$, $x_1x_200000x_9x_{10}$, $x_1x_2x_300000x_{10}$, và $x_1x_2x_3x_400000$ (lần lượt ứng với các vị trí bắt đầu dãy 00000) (**Tại sao?**)
 - Có 2^5 chuỗi dạng $00000x_6x_7x_8x_9x_{10}$
 - Với mỗi dạng còn lại, có 2^4 chuỗi
 - Do đó $|A| = 2^5 + 5 \cdot 2^4$
- Tương tự, $|B| = 2^5 + 5 \cdot 2^4$
- Tập $A \cap B$ có chính xác hai phần tử: 0000011111 và 1111100000
- Do đó, $|A \cup B| = 2(2^5 + 5 \cdot 2^4) - 2 = 222$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

17

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chồng bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Nguyên lý bù trừ



Bài tập 7

Có bao nhiêu chuỗi nhị phân độ dài 7 bắt đầu với 00 hoặc kết thúc với 111?

Bài tập 8

Một *chuỗi đối xứng (palindrome)* là một chuỗi ký tự mà khi viết ngược lại từ phải sang trái thì chuỗi không thay đổi. Có bao nhiêu chuỗi nhị phân độ dài n là chuỗi đối xứng?

Bài tập 9

Có bao nhiêu số nguyên dương nhỏ hơn hoặc bằng 1000 thỏa mãn

- (a) là bội của 7
- (b) là bội của cả 7 và 11
- (c) là bội của 7 nhưng không là bội của 11
- (d) là bội của 7 hoặc là bội của 11
- (e) không là bội của 7 và không là bội của 11

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

18 Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chồng bỏ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Nguyên lý bù trừ



Bài tập 10

Có tất cả bao nhiêu số nguyên không vượt quá 1000 là bình phương hoặc lập phương của một số nguyên dương?

Bài tập 11 (★)

Có bao nhiêu chuỗi nhị phân độ dài 8 có chứa 000 hoặc 1111?

(Đáp án: 147)

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

19 Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Quy tắc chia



Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

■ Quy tắc chia (The Division Rule):

- Một công việc có thể được thực hiện bằng n cách
- Với mỗi cách thực hiện w , có chính xác d trong n cách thực hiện tương đương/cùng loại với nó
- Số cách **khác nhau** để thực hiện công việc là n/d

■ Quy tắc chia (The Division Rule): Nếu A là hợp của m tập con đôi một không giao nhau, mỗi tập con có d phần tử, thì $m = |A|/d$

■ Quy tắc chia (The Division Rule): Nếu B là một tập hữu hạn và hàm $f : A \rightarrow B$ gán chính xác k phần tử của A cho mỗi phần tử của B , thì $|A| = k \cdot |B|$

Ví dụ 11

- Trong một đàn cừu, người ta đếm được $n = 280$ chân cừu
- Mỗi con cừu w có chính xác $d = 4$ chân
- Số con cừu trong đàn cừu là $n/d = 280/4 = 70$

20

71

Các nguyên lý đếm cơ bản

Quy tắc chia



Ví dụ 12

Có bao nhiêu cách khác nhau để sắp xếp các số 1, 1, 2?

- Đầu tiên, giả sử hai số 1 là phân biệt: 1_a và 1_b
- Có $3!$ cách sắp xếp 3 số phân biệt
- Tuy nhiên, khi coi 1_a và 1_b là cùng một số 1, *với mỗi cách sắp xếp các số 1, 1, 2, có chính xác 2 trong 3! cách "cùng loại"*, ví dụ
 - Với cách sắp xếp 1, 1, 2, có hai cách sắp xếp tương đương là $(1_a, 1_b, 2)$ và $(1_b, 1_a, 2)$
- Do đó, số cách sắp xếp khác nhau của dãy 1, 1, 2 là $3!/2 = 3$

$1_a, 1_b, 2$	$1_b, 1_a, 2$	1, 1, 2
$1_a, 2, 1_b$	$1_b, 2, 1_a$	1, 2, 1
$2, 1_a, 1_b$	$2, 1_b, 1_a$	2, 1, 1

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

21

71

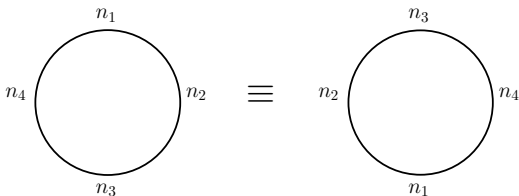
Các nguyên lý đếm cơ bản

Quy tắc chia



Ví dụ 13

Có bao nhiêu cách khác nhau để sắp xếp 4 người ngồi quanh một bàn tròn? Biết rằng *hai cách sắp xếp là giống nhau nếu mỗi người có người ngồi bên trái giống nhau và người ngồi bên phải giống nhau trong cả hai cách sắp xếp*. Ví dụ, hai cách sắp xếp sau là giống nhau



Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

22

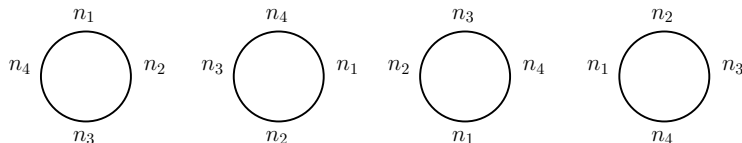
71

Các nguyên lý đếm cơ bản

Quy tắc chia



- Nếu *không có điều kiện gì* thì có tất cả $4!$ cách sắp xếp
- Chú ý rằng hai cách sắp xếp là giống nhau nếu khi ta xoay bàn sao cho n_1 nằm ở trên đỉnh thì chúng giống nhau
- Do đó, *với mỗi cách sắp xếp 4 người quanh bàn tròn, có chính xác 4 trong $4!$ cách sắp xếp "cùng loại"*
- Theo quy tắc chia, số cách khác nhau để xếp 4 người quanh bàn tròn là $4!/4 = 6$



Hình: Bốn cách sắp xếp giống nhau

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

23

Nguyên lý chuỗi bỏ câu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

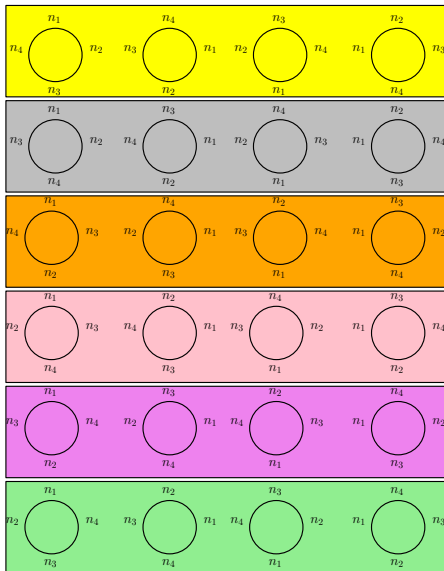
Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Quy tắc chia



Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

24

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

71

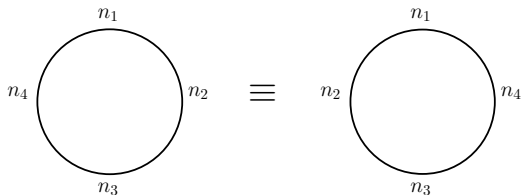
Các nguyên lý đếm cơ bản

Quy tắc chia



Bài tập 12

Giả sử hai cách sắp xếp 4 người quanh một bàn tròn là giống nhau khi mỗi người có hai người ngồi cạnh giống nhau trong cả hai cách sắp xếp không quan tâm là ngồi bên trái hay bên phải, ví dụ như hai cách sắp xếp trong hình sau là giống nhau với giả thiết hiện tại



nhưng không giống nhau với giả thiết trong Ví dụ 13. Trong trường hợp này, có bao nhiêu cách khác nhau để sắp xếp 4 người quanh một bàn tròn?

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

25

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Các nguyên lý đếm cơ bản

Quy tắc chia



Bài tập 13

Giả thiết rằng hai cách sắp xếp là giống nhau nếu mỗi người có người ngồi bên trái và người ngồi bên phải giống nhau trong mỗi cách sắp xếp

- (a) Có bao nhiêu cách khác nhau để sắp xếp 5 bạn nam và 5 bạn nữ ngồi quanh một bàn tròn?
- (b) Có bao nhiêu cách khác nhau để sắp xếp 5 bạn nam và 5 bạn nữ ngồi quanh một bàn tròn sao cho nam và nữ ngồi xen kẽ nhau?

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

26

71

Nguyên lý chuồng bồ câu



Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

27 Nguyên lý chuồng bồ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

- **Nguyên lý chuồng bồ câu (The Pigeonhole Principle)** (hay **Nguyên lý Dirichlet (The Dirichlet Drawer Principle)**): Nếu k là một số nguyên dương và có $k + 1$ con chim bồ câu hoặc nhiều hơn được đặt trong k chuồng bồ câu, thì có ít nhất một chuồng có hai con chim bồ câu hoặc nhiều hơn
- **Nguyên lý chuồng bồ câu (The Pigeonhole Principle)**: Nếu một hàm $f : A \rightarrow B$ ánh xạ một tập hữu hạn A với $|A| \geq k + 1$ đến một tập hữu hạn B với $|B| = k$, thì f không là một đơn ánh (**Nhắc lại:** f là đơn ánh khi và chỉ khi với mọi $x_1, x_2 \in A$, nếu $x_1 \neq x_2$ thì $f(x_1) \neq f(x_2)$)

Cách áp dụng nguyên lý chuồng bồ câu

- Xác định xem cái gì đại diện cho “bồ câu (pigeon)”
- Xác định xem cái gì đại diện cho “chuồng bồ câu (pigeonhole)”
- Xác định cách “bồ câu” được chia vào các “chuồng bồ câu”

Nguyên lý chuồng bồ câu



Ví dụ 14

Trong một nhóm bất kỳ có 367 người, tại sao luôn có hai người trong nhóm có cùng ngày sinh?

- Có tất cả 366 ngày sinh nhật (= “chuồng bồ câu”) và 367 người (= “bồ câu”), do đó có ít nhất hai người có cùng ngày sinh nhật

Ví dụ 15

Giả sử một kỳ thi tính các điểm số từ 0 đến 100, và mọi điểm số đều là số nguyên. Cần bao nhiêu sinh viên tham gia kỳ thi để **chắc chắn** có hai sinh viên có cùng điểm số?

- Có tất cả 101 điểm số (= “chuồng”). Cần ít nhất 102 sinh viên (= “bồ câu”) để chắc chắn có hai sinh viên có cùng điểm số

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

28 Nguyên lý chuồng bồ câu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Nguyên lý chuồng bồ câu



- **Nguyên lý chuồng bồ câu tổng quát (The Generalized Pigeonhole Principle):** Nếu N con chim bồ câu được đặt vào k chuồng bồ câu, với k là số nguyên dương nào đó, thì tồn tại một chuồng có ít nhất $\lceil N/k \rceil$ con

Ví dụ 16

- Có $N = 280$ sinh viên (= “bồ câu”) trong một lớp học. Một năm có $k = 52$ tuần (= “chuồng”). Do đó có một tuần mà ít nhất $\lceil 280/52 \rceil = \lceil 5.38 \rceil = 6$ sinh viên có ngày sinh nhật trong tuần đó
- Giá trị **lớn nhất** của d là bao nhiêu để **chắc chắn** rằng phát biểu “Trong tất cả 145 sinh viên, có ít nhất d sinh viên sinh ra vào cùng một tháng” là đúng?
 - Theo Nguyên lý chuồng bồ câu tổng quát, trong 145 sinh viên (= “bồ câu”), có ít nhất $\lceil 145/12 \rceil = 13$ sinh viên sinh ra vào cùng một tháng (= “chuồng”). Do đó, $d = 13$ **thỏa mãn yêu cầu đề ra**
 - Thêm vào đó, **phát biểu với $d \geq 14$ không đúng** vì có thể xảy ra trường hợp có 13 sinh viên sinh ra vào cùng một tháng và các tháng còn lại mỗi tháng có 12 sinh viên sinh vào tháng đó
 - Do đó $d = 13$ **là giá trị lớn nhất thỏa mãn yêu cầu đề ra**

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

29 Nguyên lý chuồng bồ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Nguyên lý chuồng bồ câu



Ví dụ 17

Chứng minh rằng trong một tập $n + 1$ số nguyên dương bất kỳ nhỏ hơn hoặc bằng $2n$, tồn tại một số là ước của một số khác trong tập đó

- Viết các số trong $n + 1$ số nguyên dương $a_1, a_2, \dots, a_n, a_{n+1}$ dưới dạng tích của một lũy thừa của 2 và một số nguyên dương lẻ, nghĩa là, $a_i = 2^{b_i} c_i$ trong đó $b_i \geq 0$ và $c_i \leq 2n$ (= “bồ câu”) là một số nguyên dương lẻ, với $1 \leq i \leq n + 1$ (**Tại sao?**)
- Có tối đa n số nguyên dương lẻ (= “chuồng”) nhỏ hơn $2n$. (**Tại sao?**) Do đó, theo nguyên lý chuồng bồ câu, có hai số c_i, c_j thỏa mãn $c_i = c_j$, với $1 \leq i, j \leq n + 1$
- Suy ra, $a_i = 2^{b_i} c_i$ và $a_j = 2^{b_j} c_j$. Do đó, nếu $b_i \leq b_j$ thì $a_i \mid a_j$ và ngược lại thì $a_j \mid a_i$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

30 Nguyên lý chuồng bồ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Nguyên lý chuồng bồ câu



Bài tập 14

Giả sử một ngăn tủ chỉ có hai loại tất màu đen và trắng, mỗi loại có 12 chiếc. Một người lấy tất trong ngăn tủ một cách ngẫu nhiên trong bóng tối

- (a) Cần lấy ít nhất bao nhiêu chiếc tất để chắc chắn có hai chiếc cùng màu? bốn chiếc cùng màu?
- (b) Cần lấy ít nhất bao nhiêu chiếc tất để chắc chắn có hai chiếc màu đen?
- (c) Nếu có thêm 12 chiếc tất màu nâu nữa trong ngăn tủ thì cần lấy ít nhất bao nhiêu chiếc tất để chắc chắn có hai chiếc cùng màu?

Bài tập 15

Chứng minh rằng trong một nhóm n số nguyên bất kỳ, có hai số nguyên có cùng số dư khi chia cho $n - 1$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

31 Nguyên lý chuồng bồ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Nguyên lý chuồng bồ câu



Bài tập 16

Có sáu giáo sư dạy toán rời rạc cơ bản tại một trường đại học. Tất cả sáu giáo sư đều sử dụng cùng một đề thi cuối kỳ. Nếu điểm thấp nhất có thể đạt được trong kỳ thi cuối kỳ là 0 và điểm cao nhất có thể là 100 và các điểm số đều là số nguyên, thì cần có ít nhất bao nhiêu sinh viên để đảm bảo rằng có hai sinh viên học cùng một giáo sư đạt được cùng một điểm thi cuối kỳ?

Bài tập 17 (★)

Chứng minh rằng với bất kỳ cách xếp 5 bạn nam và 5 bạn nữ ngồi quanh một bàn tròn, luôn tìm được một bạn (có thể là nam hoặc nữ) ngồi giữa hai bạn nam (**Gợi ý:** Chứng minh bằng phương pháp phản chứng. Giả sử có cách xếp thỏa mãn điều kiện không có bạn nào ngồi giữa hai bạn nam. Nếu chia nhóm các bạn nam thì mỗi nhóm có tối đa bao nhiêu thành viên? Giữa các nhóm này cần sắp xếp tối thiểu bao nhiêu bạn nữ để không có bạn nào ngồi giữa hai bạn nam?)

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

32 Nguyên lý chuồng bồ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Nguyên lý chuồng bồ câu



Bài tập 18

- (a) Chứng minh rằng nếu 7 số nguyên được chọn từ tập $\{1, 2, \dots, 10\}$ thì có ít nhất hai cặp trong số các số được chọn có tổng bằng 11. Nếu ta chọn 6 số nguyên thay vì 7 thì kết luận trên còn đúng không?
- (b) Cần chọn ra ít nhất bao nhiêu số từ tập $\{1, 2, 3, 4, 5, 6\}$ để chắc chắn rằng trong tập các số đã chọn có hai số có tổng bằng 7?

Bài tập 19 (*)

Chứng minh rằng trong một nhóm n người ($n \geq 2$) có ít nhất hai người có cùng số người quen biết trong nhóm

Bài tập 20

Chứng minh rằng trong một nhóm 6 người bất kỳ, luôn có ít nhất ba người đôi một biết nhau hoặc đôi một không biết nhau

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

33 Nguyên lý chuồng bồ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu



- Một **hoán vị (permutation)** của một tập S gồm các phần tử phân biệt là một dãy sắp thứ tự (ordered sequence) chứa mỗi phần tử trong S chính xác một lần
 - Tập $S = \{1, 2, 3\}$ có tất cả sáu hoán vị: $(1, 2, 3)$, $(1, 3, 2)$, $(2, 1, 3)$, $(2, 3, 1)$, $(3, 1, 2)$, $(3, 2, 1)$
- Một **chỉnh hợp chập k (k -permutation)** của một tập S là một dãy **sắp thứ tự** k phần tử **phân biệt** của S , trong đó k là một số nguyên thỏa mãn $0 \leq k \leq |S|$
 - Tập S có tất cả sáu chỉnh hợp chập 2: $(1, 2)$, $(2, 1)$, $(1, 3)$, $(3, 1)$, $(2, 3)$, $(3, 2)$
- Ký hiệu $P(n, k)$ hoặc P_n^k là **số chỉnh hợp chập k của (một tập) n phần tử**

Định lý 1

Với mọi số nguyên $n \geq 1$ và mọi số nguyên k thỏa mãn $0 \leq k \leq n$

$$P_n^k = \frac{n!}{(n-k)!}$$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

34

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu



Chứng minh Định lý 1.

Để xây dựng một chỉnh hợp chập k của một tập S gồm n phần tử, ta thực hiện một dãy k bước chọn các phần tử phân biệt trong S để xếp vào k vị trí:

- Chọn phần tử xếp vào vị trí thứ 1: có n cách chọn
- Với vị trí thứ nhất đã được xác định, chọn phần tử xếp vào vị trí thứ 2: có $n - 1$ cách chọn
- Với hai vị trí đầu đã được xác định, chọn phần tử xếp vào vị trí thứ 3: có $n - 2$ cách chọn
- ...
- Với $k - 1$ vị trí đầu đã được xác định, chọn phần tử xếp vào vị trí thứ k : có $n - k + 1$ cách chọn

Theo quy tắc nhân, có tất cả

$$n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot (n - k + 1) = \frac{n!}{(n - k)!}$$
 chỉnh hợp chập k của một tập n phần tử.

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

35

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu



Ví dụ 18

Có bao nhiêu hoán vị của các chữ cái $ABCDEFGH$ có chứa chuỗi ký tự liên tiếp ABC ?

- Số hoán vị có chứa chuỗi ký tự ABC bằng với số hoán vị của tập gồm sáu phần tử: ABC, D, E, F, G, H
- Do đó, đáp án là $6! = 720$

Ví dụ 19

Giả sử bạn cần đến 8 địa điểm khác nhau trong một thành phố nào đó. Bạn bắt buộc phải xuất phát từ một địa điểm định sẵn, nhưng có thể chọn lần lượt các địa điểm còn lại theo thứ tự bất kỳ. Có bao nhiêu thứ tự bạn có thể chọn để đến các địa điểm này?

- Địa điểm đầu tiên là cố định, còn 7 địa điểm còn lại có thể được sắp thứ tự tùy ý
- Do đó, đáp án là $7! = 5040$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

36

Giới thiệu

Một số dạng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu



Ví dụ 20

Có bao nhiêu cách khác nhau để sắp xếp 4 người ngồi quanh một bàn tròn? Biết rằng hai cách sắp xếp là giống nhau nếu mỗi người có người ngồi bên trái giống nhau và người ngồi bên phải giống nhau trong cả hai cách sắp xếp.

- Giả sử ta muốn sắp xếp n_1, n_2, n_3, n_4 quanh bàn tròn
- Để người ngồi bên trái luôn giống nhau và người ngồi bên phải luôn giống nhau trong cả hai cách sắp xếp, cách duy nhất để thu được một cách sắp xếp từ một cách khác giống nó là xoay bàn theo chiều kim đồng hồ hoặc ngược chiều kim đồng hồ.
- Do đó, để sắp xếp n_1, n_2, n_3, n_4 quanh bàn tròn:
 - Cố định vị trí của n_1 : có 1 cách (Chọn n_1 ở bất kỳ vị trí nào đều giống nhau, do nếu có một cách sắp xếp giống cách bạn sử dụng nhưng vị trí của n_1 không giống, ta có thể xoay bàn để vị trí của n_1 giống nhau trong cả hai cách sắp xếp)
 - Sắp xếp n_2, n_3, n_4 vào các vị trí còn lại: có $3! = 6$ cách
- Tóm lại, đáp án là $1 \cdot 3! = 6$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuồng bồ câu

Hoán vị, Chỉnh hợp, Tổ hợp

37

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu



- Một **tổ hợp chập k** (*k -combination*) của một tập hợp S là một dãy **không sắp thứ tự** k phần tử **phân biệt** của S , trong đó k là một số nguyên thỏa mãn $0 \leq k \leq |S|$
 - Một tổ hợp chập k của S cũng là một tập con k phần tử của S
 - Tập $S = \{1, 2, 3\}$ có ba tổ hợp chập 2: $\{1, 2\}$, $\{2, 3\}$, $\{1, 3\}$
- Ký hiệu $C(n, k)$, C_n^k , hoặc $\binom{n}{k}$ (đọc là “ n chọn k ”) là **số tổ hợp chập k của (một tập) n phần tử**

Định lý 2

Với mọi số nguyên $n \geq 1$ và mọi số nguyên k thỏa mãn $0 \leq k \leq n$

$$C_n^k = \frac{n!}{k!(n-k)!}$$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bù cầu

Hoán vị, Chỉnh hợp, Tổ hợp

38

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu



Chứng minh Định lý 2.

Để xây dựng một tổ hợp chập k của một tập S gồm n phần tử:

- Đầu tiên, ta giả thiết rằng thứ tự các phần tử là quan trọng, và xây dựng một chỉnh hợp chập k của S : có P_n^k cách
- Theo định nghĩa, trong mỗi tổ hợp chập k của S , thứ tự giữa các phần tử là không quan trọng. Nói cách khác, mỗi tổ hợp chập k của S ứng với chính xác $P_k^k = k!$ chỉnh hợp chập k có chứa cùng các phần tử và chỉ khác nhau bởi thứ tự sắp xếp các phần tử
- Theo quy tắc chia, số tổ hợp chập k của S là

$$C_n^k = \frac{P_n^k}{P_k^k} = \frac{n!}{k!(n-k)!}$$



Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuồng bồ câu

Hoán vị, Chỉnh hợp, Tổ hợp

39

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu



Ví dụ 21

Có bao nhiêu cách chọn ra 7 quân bài khác nhau từ một bộ bài 52 quân?

■ Thứ tự lựa chọn các quân bài là không quan trọng

■ Do đó, đáp án là $C_{52}^7 = \frac{52!}{7!45!} = 133\,784\,560$

Ví dụ 22

Có bao nhiêu cách chọn ra 3 bạn nam và 3 bạn nữ trong một nhóm gồm 10 bạn nam và 7 bạn nữ để đại diện tham gia một buổi họp mặt?

■ Đầu tiên ta chọn 3 bạn nam từ nhóm 10 bạn nam, sau đó chọn 3 bạn nữ từ nhóm 7 bạn nữ

■ Có C_{10}^3 cách chọn ra 3 bạn nam, và C_7^3 cách chọn ra 3 bạn nữ

■ Áp dụng quy tắc nhân, có tổng cộng $C_{10}^3 \cdot C_7^3 = 120 \cdot 35 = 4200$ cách

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bồ câu

Hoán vị, Chỉnh hợp, Tổ hợp

40

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Hoán vị, Chính hợp, Tổ hợp

Một số đẳng thức tổ hợp



Định lý 3

Với các số nguyên n, k thỏa mãn $0 \leq k \leq n$, ta có

$$C_n^k = C_n^{n-k}$$

Chứng minh.

- Giả sử $\mathcal{A}$ là tập các tập con k phần tử và $\mathcal{B}$ là tập các tập con $n - k$ phần tử của một tập n phần tử X . Hàm $f: \mathcal{A} \rightarrow \mathcal{B}$ cho bởi $f(A) = X \setminus A$ với $A \in \mathcal{A}$ là một song ánh (**Tại sao?**)
- Do đó, $C_n^k = C_n^{n-k}$



Bài tập 21

Chứng minh Định lý 3 bằng công thức đã biết.

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuồng bồ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

41

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Hoán vị, Chỉnh hợp, Tổ hợp

Một số đẳng thức tổ hợp



Ta có thể chứng minh hai vế của một đẳng thức bằng cách chỉ ra chúng *đếm cùng một đối tượng thông qua các phương pháp khác nhau*. Phương pháp chứng minh này gọi là *phương pháp đếm bằng hai cách (double counting proof)*

Trong Ví dụ 23, với $n, k \geq 2$ và $n \geq k + 2$, ta sử dụng phương pháp đếm bằng hai cách để chứng minh đẳng thức

$$C_n^k - C_{n-2}^{k-2} = 2C_{n-2}^{k-1} + C_{n-2}^k$$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuồng bồ câu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

42 Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Hoán vị, Chỉnh hợp, Tổ hợp

Một số đẳng thức tổ hợp



Ví dụ 23 (Đếm bằng hai cách)

Giả sử $n, k \geq 2$ và $n \geq k + 2$. Ta đếm số cách chọn k số nguyên từ tập $\{1, 2, \dots, n\}$ sao cho 1 và 2 không đồng thời được chọn

■ Cách 1:

- Có C_n^k tập con k phần tử và không có hạn chế gì
- Có C_{n-2}^{k-2} tập con k phần tử có chứa đồng thời cả 1 và 2
 - Cố định 1 và 2, chọn $k - 2$ phần tử còn lại từ tập $\{3, \dots, n\}$
- Do đó đáp án là $C_n^k - C_{n-2}^{k-2}$

■ Cách 2:

- Có C_{n-2}^{k-1} tập con k phần tử chứa 1 nhưng không chứa 2
 - Cố định 1, chọn $k - 1$ phần tử còn lại từ tập $\{3, \dots, n\}$
- Tương tự, có C_{n-2}^{k-1} tập con k phần tử chứa 2 nhưng không chứa 1
- Có C_{n-2}^k tập con k phần tử không chứa cả 1 và 2
 - Chọn k phần tử từ tập $\{3, \dots, n\}$
- Do đó đáp án là $2C_{n-2}^{k-1} + C_{n-2}^k$

Do đó, $C_n^k - C_{n-2}^{k-2} = 2C_{n-2}^{k-1} + C_{n-2}^k$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

43

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Hoán vị, Chỉnh hợp, Tổ hợp

Một số đẳng thức tổ hợp



Định lý 4

Giả sử $n > k \geq 1$. Ta có $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$

Chứng minh.

- C_n^k là số cách chọn k phần tử từ tập n phần tử $\{1, 2, \dots, n\}$
- Để chọn k phần tử từ $\{1, 2, \dots, n\}$, ta cũng có thể:
 - Chọn 1, và chọn $k - 1$ phần tử còn lại từ tập $\{2, \dots, n\}$: có C_{n-1}^{k-1} cách chọn
 - Không chọn 1, và chọn k phần tử từ tập $\{2, \dots, n\}$: có C_{n-1}^k cách chọn

Do đó, theo quy tắc cộng, có $C_{n-1}^k + C_{n-1}^{k-1}$ tập con k phần tử của tập n phần tử

- Do đó, $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$



Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

44

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Hoán vị, Chỉnh hợp, Tổ hợp

Một số đẳng thức tổ hợp



Bài tập 22

Chứng minh Định lý 4 bằng công thức đã biết

Bài tập 23

Sử dụng Định lý 4, chứng minh đẳng thức thu được từ Ví dụ 23 sau

$$C_n^k - C_{n-2}^{k-2} = 2C_{n-2}^{k-1} + C_{n-2}^k$$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuồng bồ câu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

45

Hoán vị, Chỉnh hợp, Tổ hợp

Một số đẳng thức tổ hợp



Bài tập 24

(a) Chứng minh rằng với mọi $n > 0$, $\sum_{k=0}^n (-1)^k C_n^k = 0$

(b) Chứng minh bằng phương pháp quy nạp rằng với mọi

$$m > 0, \sum_{k=1}^m (-1)^{k-1} C_m^k = 1. \text{ (Gợi ý: } C_n^k = C_{n-1}^k + C_{n-1}^{k-1} \text{)}$$

Bài tập 25

Chứng minh **đẳng thức Vandermonde (Vandermonde's Identity)** sau

$$C_{m+n}^k = \sum_{i=0}^k C_m^{k-i} C_n^i,$$

trong đó m, n, k là các số nguyên không âm và $k \leq \min(m, n)$.
(**Gợi ý:** Có bao nhiêu cách chọn ra một ban đại diện gồm k thành viên từ một lớp học có m nam và n nữ?)

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

46

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

71

Hoán vị, Chỉnh hợp, Tổ hợp

Một số đẳng thức tổ hợp



Bài tập 26

Chứng minh rằng với mọi $n \geq 0$, $\sum_{k=0}^n (C_n^k)^2 = C_{2n}^n$. (**Gợi ý:** Có bao nhiêu chuỗi nhị phân độ dài $2n$ có chính xác n số 0?)

Bài tập 27

Chứng minh rằng với mọi $n \geq 3$, $\sum_{k=2}^{n-1} C_k^2 = C_n^3$. (**Gợi ý:** Có bao nhiêu cách chọn một tập con gồm 3 phần tử a, b, c từ tập $\{1, \dots, n\}$ với $a < b < c$? Xét các trường hợp có thể xảy ra với a)

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuồng bồ câu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

47

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal



Bắt đầu với $n = 0$, hàng thứ n có $n + 1$ phần tử: $C_n^0, C_n^1, \dots, C_n^n$

$$C_0^0 = 1$$

$$C_1^0 = 1 \quad C_1^1 = 1$$

$$C_2^0 = 1 \quad C_2^1 = 2 \quad C_2^2 = 1$$

$$C_3^0 = 1 \quad C_3^1 = 3 \quad C_3^2 = 3 \quad C_3^3 = 1$$

$$C_4^0 = 1 \quad C_4^1 = 4 \quad C_4^2 = 6 \quad C_4^3 = 4 \quad C_4^4 = 1$$

Hình: Tam giác Pascal

Định lý 3 và 4 có thể được minh họa bằng tam giác Pascal

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

48

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal



Một tính chất khác của tam giác Pascal là

Định lý 5

Với mọi $n \geq 0$, tổng các phần tử ở hàng thứ n là

$$\sum_{k=0}^n C_n^k = 2^n$$

Chứng minh.

- C_n^k là số cách chọn tập con k phần tử từ một tập n phần tử
- Do đó, $\sum_{k=0}^n C_n^k$ là số tập con của một tập n phần tử
- Như ta đã đề cập trong phần quy tắc nhân, số tập con của một tập n phần tử là 2^n

Bài tập 28

Chứng minh Định lý 5 bằng phương pháp quy nạp. (Gợi ý: Sử dụng Định lý 4)

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

49

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Tam giác Pascal và Định lý nhị thức

Định lý nhị thức



Bài toán: Khai triển $(x + y)^n$

Ví dụ 24

$$(x + y)^0 = 1$$

$$(x + y)^1 = 1 \cdot x + 1 \cdot y$$

$$(x + y)^2 = 1 \cdot x^2 + 2 \cdot xy + 1 \cdot y^2$$

$$(x + y)^3 = 1 \cdot x^3 + 3 \cdot x^2y + 3 \cdot xy^2 + 1 \cdot y^3$$

$$(x + y)^4 = 1 \cdot x^4 + 4 \cdot x^3y + 6 \cdot x^2y^2 + 4 \cdot xy^3 + 1 \cdot y^4$$

Nhận xét: Các hệ số khi khai triển $(x + y)^n$ ($0 \leq n \leq 4$) giống với các hàng tương ứng trong tam giác Pascal

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

50

71

Tam giác Pascal và Định lý nhị thức

Định lý nhị thức



Định lý 6: Định lý nhị thức

$$(x + y)^n = \sum_{k=0}^n C_n^k x^{n-k} y^k$$

Chứng minh.

- Để nhận được một số hạng $x^{n-k}y^k$ trong khai triển của $(x + y)^n = (x + y)(x + y) \dots (x + y)$, ta cần chọn chính xác $n - k$ số x và k số y từ n thừa số $(x + y)$. (Với mỗi thừa số, chọn chính xác một phần tử: hoặc x hoặc y)
 - Có C_n^{n-k} cách chọn chính xác $n - k$ số x
 - Ứng với mỗi cách chọn $n - k$ số x , có chính xác 1 cách chọn k số y từ k thừa số $(x + y)$ còn lại
- Do đó, hệ số của $x^{n-k}y^k$ là $C_n^{n-k} = C_n^k$

Bài tập 29

Chứng minh Định lý 5 bằng cách sử dụng Định lý 6

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chồng bỏ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

51

71

Tam giác Pascal và Định lý nhị thức

Định lý nhị thức



Ví dụ 25

Khai triển $(x + 2)^4$

$$\begin{aligned}(x + 2)^4 &= C_4^0 \cdot x^4 + C_4^2 \cdot x^3 \cdot 2^1 + C_4^3 \cdot x^2 \cdot 2^2 + \\ &\quad + C_4^1 \cdot x^1 \cdot 2^3 + C_4^4 \cdot x^0 \cdot y^4 \\ &= x^4 + 8x^3 + 24x^2 + 32x + 16\end{aligned}$$

Ví dụ 26

Tính $(1.02)^7$ làm tròn đến chữ số thập phân thứ 4

$$\begin{aligned}(1 + 0.02)^7 &= C_7^0 \cdot 1^7 + C_7^1 \cdot 1^6 \cdot (0.02) + C_7^2 \cdot 1^5 \cdot (0.0004) \\ &\quad + C_7^3 \cdot 1^4 \cdot (0.000008) + \dots \\ &= 1 + 1.14 + 0.0084 + 0.00028 + \dots \\ &\approx 1.14868 \approx 1.1487\end{aligned}$$

Chú ý: Để làm tròn đến chữ số thập phân thứ 4, ta cần tính toán đến chữ số thập phân thứ 5

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

52

71

Tam giác Pascal và Định lý nhị thức

Định lý nhị thức



Bài tập 30

Chứng minh rằng với mọi $n \geq 0$, $\sum_{k=0}^n 2^k C_n^k = 3^n$

- (a) bằng cách áp dụng Định lý nhị thức (Định lý 6)
- (b) (*) bằng phương pháp đếm hai lần (**Gợi ý:** Có bao nhiêu cách xây dựng một chuỗi n ký tự chỉ sử dụng các ký tự A, B, C thỏa mãn điều kiện có chính xác $n - k$ ký tự A ?)

Bài tập 31

Tìm hệ số

- (a) của x^7 trong khai triển của $(1 + x)^{11}$
- (b) của x^9 trong khai triển của $(2 - x)^{19}$
- (c) (*) của x^k trong khai triển của $\left(x + \frac{1}{x}\right)^{100}$, trong đó k là một số nguyên

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

53

71

Tam giác Pascal và Định lý nhị thức

Nguyên lý bù trừ tổng quát



Định lý 7: Nguyên lý bù trừ tổng quát

Với các tập hữu hạn $A_1, A_2, \dots, A_n$

$$\begin{aligned} \left| \bigcup_{k=1}^n A_k \right| &= \sum_{k=1}^n \left(\sum_{\{I \mid I \subseteq \{1, \dots, n\} \wedge |I|=k\}} (-1)^{k-1} \left| \bigcap_{i \in I} A_i \right| \right) \\ &= (-1)^0 |A_1| + (-1)^0 |A_2| + \dots + (-1)^0 |A_n| \\ &\quad + (-1)^1 |A_1 \cap A_2| + \dots + (-1)^1 |A_{n-1} \cap A_n| \\ &\quad + \dots \\ &\quad + (-1)^{n-1} |A_1 \cap A_2 \cap \dots \cap A_n| \end{aligned}$$

Bài tập 32

- (a) Khai triển $|A_1 \cup \dots \cup A_n|$ theo nguyên lý bù trừ tổng quát, với $n \in \{1, 2, 3, 4\}$
- (b) Theo nguyên lý bù trừ tổng quát, $|A_1 \cup \dots \cup A_n|$ được khai triển thành tổng của bao nhiêu số hạng?

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

54 Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

Một số ví dụ khác

Tam giác Pascal và Định lý nhị thức

Nguyên lý bù trừ tổng quát



Bài tập 33

Có 345 sinh viên tại một trường đại học đã học khóa giải tích, 212 sinh viên đã học khóa toán rời rạc, và 188 sinh viên đã học cả hai khóa giải tích và toán rời rạc. Hỏi có bao nhiêu sinh viên đã học ít nhất một trong hai khóa giải tích hoặc toán rời rạc?

Bài tập 34

Một khảo sát các hộ gia đình ở Hoa Kỳ cho thấy 96% có ít nhất một chiếc tivi, 98% có dịch vụ điện thoại, và 95% có cả dịch vụ điện thoại và ít nhất một chiếc tivi. Có bao nhiêu phần trăm hộ gia đình ở Hoa Kỳ không có cả dịch vụ điện thoại lẫn tivi?

Bài tập 35

Một báo cáo tiếp thị về máy tính cá nhân cho biết 650 000 chủ sở hữu sẽ mua một máy in cho máy của họ trong năm tới và 1 250 000 sẽ mua ít nhất một gói phần mềm. Nếu báo cáo cho biết 1 450 000 chủ sở hữu sẽ mua máy in hoặc ít nhất một gói phần mềm, hỏi có bao nhiêu người sẽ mua cả máy in và ít nhất một gói phần mềm?

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

55

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Tam giác Pascal và Định lý nhị thức

Nguyên lý bù trừ tổng quát



Bài tập 36

- (a) Tìm số các số nguyên dương không vượt quá 100 mà không chia hết cho 5 và cũng không chia hết cho 7.
- (b) Tìm số các số nguyên dương không vượt quá 1000 mà không chia hết cho 3, 17 hoặc 35.
- (c) Tìm số các số nguyên dương không vượt quá 10 000 mà không chia hết cho 3, 4, 7 hoặc 11.

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuồng bồ câu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

56

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Chỉnh hợp và tổ hợp suy rộng



- Một **chỉnh hợp lặp chập k** (*k -permutation with repetition*) của một tập S là một dãy **sắp thứ tự** k phần tử của S trong đó các phần tử **có thể lặp lại**
- Một **tổ hợp lặp chập k của một tập S** (*k -combination with repetition*) của một tập S là một dãy **không sắp thứ tự** k phần tử của S trong đó các phần tử **có thể lặp lại**

Ví dụ 27

Xét tập $S = \{a, b, c, d, e\}$ với $|S| = 5$.

- (a, a, b, c, d, e) , (a, b, a, c, d, e) , và (e, e, e, e, e, e) đều là các chỉnh hợp lặp chập 6 của S
- $\{a, a, b, c, d, e\}$, $\{a, b, a, c, d, e\}$, và $\{e, e, e, e, e, e\}$ đều là các tổ hợp lặp chập 6 của S
- Các dãy $\{a, a, b, c, d, e\}$, $\{e, d, c, b, a, a\}$, $\{a, e, a, d, c, b\}$, ... đều biểu diễn cùng một tổ hợp lặp $\{a, a, b, c, d, e\}$ (có phần tử a xuất hiện hai lần và các phần tử b, c, d, e xuất hiện một lần)

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

57

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Chỉnh hợp và tổ hợp suy rộng



Định lý 8

Số chỉnh hợp lặp chập k của một tập n phần tử là n^k

Chứng minh.

Một chỉnh hợp lặp chập k của một tập n phần tử có thể được xây dựng như sau:

- Có n lựa chọn cho vị trí thứ nhất
- Với mỗi lựa chọn cho vị trí thứ nhất, có n lựa chọn cho vị trí thứ hai
- ...
- Với mỗi lựa chọn cho các vị trí thứ nhất, thứ hai, ..., thứ $(k - 1)$, có n lựa chọn cho vị trí thứ k
- Theo quy tắc nhân, có n^k cách xây dựng một chỉnh hợp lặp chập k của một tập n phần tử

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

58

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Chỉnh hợp và tổ hợp suy rộng



Định lý 9

Số tổ hợp lặp chập k của một tập n phần tử là C_{n+k-1}^{n-1}

Chứng minh.

- Mỗi tổ hợp lặp chập k từ tập n phần tử có thể biểu diễn bằng một dãy $n - 1$ thanh đứng và k ngôi sao
 - Ta dùng $n - 1$ thanh đứng để phân cách các ngăn
 - Ngăn thứ i chứa thêm một ngôi sao mỗi lần khi phần tử thứ i của tập xuất hiện trong tổ hợp
 - Ví dụ, một tổ hợp lặp chập 6 của tập 4 phần tử được biểu diễn bởi

$** \mid * \mid \mid * * *$

là một tổ hợp chứa đúng hai phần tử thứ nhất, một phần tử thứ hai, không phần tử thứ ba, và ba phần tử thứ tư của tập 4 phần tử

- Để xây dựng một dãy $n - 1$ thanh đứng và k ngôi sao, trước tiên ta chọn vị trí cho k ngôi sao trong dãy $n + k - 1$ ký tự và sau đó điền các thanh đứng vào $n - 1$ vị trí còn lại
 - Số cách chọn vị trí cho k ngôi sao trong dãy $n + k - 1$ ký tự là C_{n+k-1}^k
 - Số cách điền các thanh đứng vào các vị trí còn lại là 1
 - Do đó, số các dãy thanh đứng và ngôi sao là $C_{n+k-1}^k = C_{n+k-1}^{n-1}$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chồng bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

59

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Chỉnh hợp và tổ hợp suy rộng



Ví dụ 28

Có bao nhiêu cách để 6 quả bóng vào một túi, biết rằng mỗi quả bóng chỉ có thể có màu đỏ (R), xanh lá cây (G), hoặc xanh da trời (B)?

- 6 quả bóng ứng với 6 ngôi sao, và 2 thanh đứng phân cách thành ba ngăn ứng với các màu đỏ, xanh lá cây, xanh da trời
 - Lựa chọn [R, R, G, G, G, B] ứng với dãy ** | * * * | *
 - Lựa chọn [R, B, R, R, B, R] ứng với dãy * * * * || **
- Số cách để 6 quả bóng vào túi là $C_{3+6-1}^{3-1} = 28$

Ví dụ 29

Phương trình $x_1 + x_2 + x_3 = 11$ có bao nhiêu nghiệm nguyên thỏa mãn $x_1 \geq 0$, $x_2 \geq 0$, và $x_3 \geq 0$?

- Mỗi nghiệm của phương trình ứng với một cách để 11 quả bóng vào 3 hộp gán nhãn x_1, x_2, x_3
- Số nghiệm nguyên không âm của phương trình chính là tổ hợp lặp chập 11 của tập 3 phần tử: $C_{3+11-1}^{3-1} = 78$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

60

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

71

Chỉnh hợp và tổ hợp suy rộng



Định lý 10

Cho phương trình $x_1 + x_2 + \dots + x_n = k$ với $n \geq 1$ và $k \geq 0$, số nghiệm nguyên không âm của phương trình này (hay số bộ $(x_1, x_2, \dots, x_n)$ thỏa mãn phương trình, trong đó $x_i \in \mathbb{N}$) là C_{n+k-1}^{n-1}

- Mỗi nghiệm của phương trình ứng với một cách để k quả bóng vào n hộp gán nhãn $x_1, x_2, \dots, x_n$
- Số nghiệm nguyên không âm của phương trình chính là tổ hợp lặp chập k của tập n phần tử: C_{n+k-1}^{n-1}
- **Chú ý:** $x_1, x_2, \dots, x_n$ là các **số nguyên không âm**

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

61 Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Chỉnh hợp và tổ hợp suy rộng



Ví dụ 30

Phương trình $x_1 + x_2 + x_3 = 11$ có bao nhiêu nghiệm nguyên thỏa mãn $x_1 \geq -2$, $x_2 \geq 1$, và $x_3 \geq 0$?

- Ta *xây dựng một phương trình mới có dạng tổng của các số nguyên không âm* như ở ví dụ trước
 - Ta có $x_1 + 2 \geq 0$ và $x_2 - 1 \geq 0$
 - Đặt $x'_1 = x_1 + 2 \geq 0$ và $x'_2 = x_2 - 1 \geq 0$
 - Số nghiệm nguyên của phương trình $x_1 + x_2 + x_3 = 11$ thỏa mãn $x_1 \geq -2$, $x_2 \geq 1$, và $x_3 \geq 0$ cũng là số nghiệm nguyên không âm của phương trình $x'_1 + x'_2 + x_3 = 11 + 2 - 1 = 12$
- Số nghiệm thỏa mãn điều kiện là $C_{3+12-1}^{3-1} = 91$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

62

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Chỉnh hợp và tổ hợp suy rộng



Ví dụ 31

Phương trình $x_1 + x_2 + x_3 = 11$ có bao nhiêu nghiệm nguyên thỏa mãn $0 \leq x_1 \leq 2, x_2 \geq 0, x_3 \geq 0$?

■ Ta đếm số nghiệm thỏa mãn

■ $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$

■ $x_1 \geq 3, x_2 \geq 0, x_3 \geq 0$

Số nghiệm thỏa mãn $0 \leq x_1 \leq 2, x_2 \geq 0, x_3 \geq 0$ sẽ là hiệu của hai số trên

■ Có $C_{3+11-1}^{11} = 78$ nghiệm nguyên thỏa mãn $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$. Có $C_{3+8-1}^8 = 45$ nghiệm nguyên thỏa mãn $x_1 \geq 3, x_2 \geq 0, x_3 \geq 0$. Do đó, kết quả là $C_{3+11-1}^{11} - C_{3+8-1}^8 = 78 - 45 = 33$

Bài tập 37

Đếm số nghiệm nguyên của $x_1 + x_2 + x_3 = 11$ thỏa mãn $0 \leq x_1 \leq 2, x_2 \geq 0, x_3 \geq 0$ một cách trực tiếp bằng cách xét từng trường hợp $x_1 = 0, x_1 = 1$, và $x_1 = 2$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

63 Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Chỉnh hợp và tổ hợp suy rộng



Định lý 11

Số hoán vị phân biệt của n phần tử trong đó có n_1 phần tử giống nhau thuộc loại 1, n_2 phần tử giống nhau thuộc loại 2, ..., và n_k phần tử giống nhau thuộc loại k , là
$$\frac{n!}{n_1!n_2!\dots n_k!}$$

Định lý 12

Số cách chia n vật khác nhau vào trong k hộp sao cho có n_i đồ vật được đặt vào hộp thứ i , với $i = 1, 2, \dots, k$ là

$$\frac{n!}{n_1!n_2!\dots n_k!}$$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

64

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

Chỉnh hợp và tổ hợp suy rộng



Ví dụ 32

Có bao nhiêu chuỗi ký tự phân biệt thu được bằng cách sắp xếp lại thứ tự các chữ cái trong chuỗi **SUCCESS**?

- Số chuỗi ký tự chính là số hoán vị phân biệt của 7 ký tự, trong đó có 3 ký tự **S**, 2 ký tự **C**, 1 ký tự **U**, và 1 ký tự **E**
- Do đó, có tất cả $\frac{7!}{3!2!1!1!} = 420$ chuỗi ký tự phân biệt

Ví dụ 33

Có bao nhiêu cách chia 5 quân bài cho mỗi người trong số 4 người chơi từ một bộ bài 52 quân thông thường?

- Mỗi cách chia bài ứng với một cách bỏ 52 quân bài vào 5 “hộp” phân biệt (4 “hộp” ứng với 4 người chơi và 1 “hộp” ứng với các quân bài không được chia cho ai cả)
- Do đó, có tất cả $\frac{52!}{5!5!5!32!}$ cách chia bài

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

65

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

71

Chỉnh hợp và tổ hợp suy rộng



Hệ số đa thức tổng quát

Với các số nguyên không âm $n, n_1, n_2, \dots, n_k$ thỏa mãn $n = n_1 + n_2 + \dots + n_k$, ta định nghĩa

$$C_n^{n_1, n_2, \dots, n_k} = \frac{n!}{n_1! n_2! \dots n_k!}$$

Định lý 13: Định lý đa thức

Với mọi $n \geq 0$ và $k \geq 1$

$$(x_1 + \dots + x_k)^n = \sum_{\substack{0 \leq n_1, \dots, n_k \leq n \\ n_1 + \dots + n_k = n}} C_n^{n_1, n_2, \dots, n_k} x_1^{n_1} \dots x_k^{n_k}$$

- Định lý nhị thức là trường hợp đặc biệt của định lý đa thức với $k = 2$

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

66

Chỉnh hợp và tổ hợp suy rộng

Một số ví dụ khác

71

Một số ví dụ khác



Ví dụ 34

Có bao nhiêu cách phân phối 4 nhân viên khác nhau A, B, C, D vào 3 văn phòng hoàn toàn giống hệt nhau? (Chú ý là xếp A, B vào phòng thứ nhất và C, D vào phòng thứ hai hoàn toàn giống với việc xếp A, B vào phòng thứ hai và C, D vào phòng thứ ba. *Điều quan trọng là A, B cùng phòng và C, D cùng phòng.*) Giả sử rằng mỗi văn phòng có thể chứa được bất kỳ một số lượng nhân viên nào

- Cả 4 người chung một văn phòng: có $C_4^4 = 1$ cách
 - Xếp 4 người vào bất kỳ phòng nào đều được
- ba + một: có $C_4^3 = 4$ cách
 - Chọn 3 người xếp vào một phòng, người còn lại tự động xếp vào một trong hai phòng còn lại, phòng nào đều được
- hai + hai: có $C_4^2/2 = 3$ cách
 - Chọn 2 người để xếp vào một phòng, hai người còn lại tự động vào một trong hai phòng còn lại. Mỗi cách chọn này có một cách tương đương với nó, ví dụ như chọn A, B xếp vào một phòng tương đương với chọn C, D xếp vào một phòng, vì đều cho kết quả là $\{\{A, B\}, \{C, D\}\}$
- hai + một + một: có C_4^2 cách
 - Chọn 2 người để xếp vào một phòng, hai người còn lại tự động xếp vào hai phòng còn lại

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

67 Một số ví dụ khác

Một số ví dụ khác



Ví dụ 35

Có bao nhiêu cách để đặt 6 quyển sách hoàn toàn giống nhau vào 4 hộp hoàn toàn giống nhau, trong đó mỗi hộp có thể chứa nhiều nhất 6 quyển sách?

- Ta liệt kê các cách sắp xếp bằng cách liệt kê số sách lớn nhất trong một hộp, theo sau bởi các số sách nhỏ hơn trong các hộp có chứa ít nhất một quyển sách khác, theo thứ tự giảm dần của số sách. Ví dụ, 4, 1, 1 mô tả cách xếp sách vào 3 hộp, một hộp có 4 quyển, hai hộp khác mỗi hộp có 1 quyển
- Các cách sắp xếp là:
 - 6
 - 5, 1
 - 4, 2 4, 1, 1
 - 3, 3 3, 2, 1 3, 1, 1, 1
 - 2, 2, 2 2, 2, 1, 1

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số dạng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

68

Một số ví dụ khác

71

Một số ví dụ khác

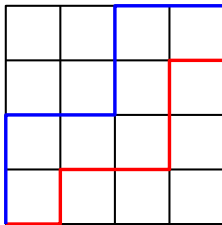


Bài tập 38 (★)

Alice và Bob chơi trò chơi sau: Bob chọn 10 số nguyên bất kỳ trong khoảng từ 1 đến 40. Alice cần tìm hai tập số nguyên khác nhau trong các số mà Bob chọn, mỗi tập có 3 phần tử, sao cho tổng các số nguyên trong hai tập là bằng nhau. Hãy chứng minh rằng Alice luôn luôn thắng. (**Gợi ý:** Áp dụng nguyên lý chuồng bồ câu)

Bài tập 39

Có bao nhiêu cách đi từ góc dưới cùng bên trái đến góc trên cùng bên phải của một lưới kích thước $n \times n$? Giả sử rằng trong mỗi bước từ một đỉnh sang đỉnh khác của lưới, bạn chỉ có thể đi sang phải một bước hoặc đi lên trên một bước



Hình: Một lưới 4×4 và ví dụ một số đường đi

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuồng bồ câu

Hoán vị, Chính hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chính hợp và tổ hợp suy rộng

69

Một số ví dụ khác

71

Một số ví dụ khác



Bài tập 40

Có bao nhiêu chuỗi nhị phân độ dài n không có hai số 1 liên tiếp? (**Gợi ý:** Gọi a_n là số chuỗi nhị phân độ dài n không có hai số 1 liên tiếp. Tìm và giải hệ thức truy hồi cho a_n)

Bài tập 41

Bất phương trình $x_1 + x_2 + x_3 \leq 11$ có bao nhiêu nghiệm nguyên thỏa mãn $x_1 \geq 0$, $x_2 \geq 0$, và $x_3 \geq 0$? (**Gợi ý:** Đưa vào thêm một biến tạm thời x_4 thỏa mãn $x_1 + x_2 + x_3 + x_4 = 11$.)

Bài tập 42

Phương trình $x_1 + x_2 + x_3 + x_4 + x_5 = 21$ có bao nhiêu nghiệm nguyên thỏa mãn điều kiện $x_i \geq 0$ với mọi $i \in \{1, 2, 3, 4, 5\}$ và

- (a) $x_1 \geq 1$?
- (b) $x_i \geq 2$ với mọi $i \in \{1, 2, 3, 4, 5\}$?
- (c) $0 \leq x_1 \leq 10$?
- (d) (*) $0 \leq x_1 \leq 3$, $1 \leq x_2 < 4$, và $x_3 \geq 15$?

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bỏ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

70

Một số ví dụ khác

71



Một số ví dụ khác

Bài tập 43 (★)

Có bao nhiêu số nguyên dương nhỏ hơn 1 000 000 có tổng các chữ số bằng 19?

Bài tập 44

Chứng minh rằng nếu n và k là các số nguyên với $1 \leq k \leq n$ thì

$$kC_n^k = nC_{n-1}^{k-1} \quad (1)$$

- (a) bằng cách sử dụng các công thức tổ hợp và biến đổi đại số
- (b) bằng phương pháp đếm hai lần

Bài tập 45

Chứng minh rằng nếu n là một số nguyên dương thì

$$C_{2n}^2 = 2C_n^2 + n^2 \quad (2)$$

- (a) bằng cách sử dụng các công thức tổ hợp và biến đổi đại số
- (b) bằng phương pháp đếm hai lần

Các phương pháp đếm

Hoàng Anh Đức

Các nguyên lý đếm cơ bản

Quy tắc song ánh

Quy tắc nhân và Quy tắc cộng

Nguyên lý bù trừ

Quy tắc chia

Nguyên lý chuỗi bổ cầu

Hoán vị, Chỉnh hợp, Tổ hợp

Giới thiệu

Một số đẳng thức tổ hợp

Tam giác Pascal và Định lý nhị thức

Tam giác Pascal

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Chỉnh hợp và tổ hợp suy rộng

71 Một số ví dụ khác

Part I

Phụ lục

Nội dung



Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Hàm sinh

Một số lỗi thường gặp

Tam giác Pascal và Định lý nhị thức
Nguyên lý bù trừ tổng quát

Hàm sinh

Một số lỗi thường gặp

Tam giác Pascal và Định lý nhị thức

Nguyên lý bù trừ tổng quát



Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và
Định lý nhị thức

2 Nguyên lý bù trừ tổng quát

Hàm sinh

Một số lỗi thường gặp

Định lý 14: Nguyên lý bù trừ tổng quát

Với các tập hữu hạn $A_1, A_2, \dots, A_n$

$$\begin{aligned} \left| \bigcup_{k=1}^n A_k \right| &= \sum_{k=1}^n \left(\sum_{\{I \mid I \subseteq \{1, \dots, n\} \wedge |I|=k\}} (-1)^{k-1} \left| \bigcap_{i \in I} A_i \right| \right) \\ &= (-1)^0 |A_1| + (-1)^0 |A_2| + \dots + (-1)^0 |A_n| \\ &\quad + (-1)^1 |A_1 \cap A_2| + \dots + (-1)^1 |A_{n-1} \cap A_n| \\ &\quad + \dots \\ &\quad + (-1)^{n-1} |A_1 \cap A_2 \cap \dots \cap A_n| \end{aligned}$$

Tam giác Pascal và Định lý nhị thức

Nguyên lý bù trừ tổng quát



Chứng minh.

$$\left| \bigcup_{k=1}^n A_k \right| = \sum_{k=1}^n \left(\sum_{\{I | I \subseteq \{1, \dots, n\} \wedge |I|=k\}} (-1)^{k-1} \left| \bigcap_{i \in I} A_i \right| \right)$$

- Cho $a \in \bigcup_{k=1}^n A_k$ và giả thiết rằng a xuất hiện trong chính xác m tập $A_{a_1}, A_{a_2}, \dots, A_{a_m}$, với $1 \leq a_1, a_2, \dots, a_m \leq n$
- a được đếm một lần ở về trái
- Có bao nhiêu lần a được đếm ở về phải?
 - a xuất hiện trong C_m^1 tập $A_{a_1}, A_{a_2}, \dots, A_{a_m}$
 - a xuất hiện trong C_m^2 tập $A_{a_i} \cap A_{a_j}$ với $1 \leq i < j \leq m$
 - a xuất hiện trong C_m^3 tập $A_{a_i} \cap A_{a_j} \cap A_{a_k}$ với $1 \leq i < j < k \leq m$
 - ...

Do đó, ở về phải, a được đếm $\sum_{k=1}^m (-1)^{k-1} C_m^k$ lần

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và
Định lý nhị thức

Nguyên lý bù trừ tổng quát

Hàm sinh

Một số lỗi thường gặp

3

Tam giác Pascal và Định lý nhị thức

Nguyên lý bù trừ tổng quát



Chứng minh (tiếp).

- Để chứng minh vế trái bằng vế phải, ta cần chỉ ra a cũng được đếm một lần ở vế phải, nghĩa là

$$\sum_{k=1}^m (-1)^{k-1} C_m^k = 1$$

- Theo Định lý 6, ta có

$$\begin{aligned} 0 &= (-1 + 1)^m \\ &= \sum_{k=0}^m C_m^k (-1)^k 1^{m-k} \\ &= 1 + \sum_{k=1}^m (-1)^k C_m^k \end{aligned}$$

$$\text{Do đó, } \sum_{k=1}^m (-1)^k C_m^k = -1. \text{ Suy ra } \sum_{k=1}^m (-1)^{k-1} C_m^k = 1$$

(Chia hai vế cho -1)

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và
Định lý nhị thức

4 Nguyên lý bù trừ tổng quát

Hàm sinh

Một số lỗi thường gặp

Hàm sinh



Hàm sinh

Hàm sinh ((ordinary) generating function) $G(x)$ của một *dãy vô hạn* $\{a_n\}$ ($n \geq 0$) được định nghĩa bởi $G(x) = \sum_{n=0}^{\infty} a_n x^n$.

Nói cách khác, a_n là hệ số của x^n trong $G(x)$

- Để thuận tiện, ta viết dãy $\{a_n\} = \langle a_0, a_1, a_2, a_3, \dots \rangle$
- Lưu ý rằng biến x trong hàm sinh không biểu diễn bất kỳ đại lượng nào mà chỉ đóng vai trò như một ký hiệu để thuận tiện cho việc đề cập đến “hệ số của x^n ”

Ví dụ 36

(a) Hàm sinh của dãy $\langle 1, 1, 1, 1, \dots \rangle$ là $G(x) = \sum_{n=0}^{\infty} x^n$

(b) Hàm sinh của dãy $\langle 1, -1, 1, -1, \dots \rangle$ là $G(x) = \sum_{n=0}^{\infty} (-1)^n x^n$

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

5 Hàm sinh

Một số lỗi thường gặp



Hàm sinh

Hàm sinh ((ordinary) generating function) $G(x)$ của một lớp (tập hợp) các đối tượng $\mathcal{A}$ là $G(x) = \sum_{n=0}^{\infty} a_n x^n$, trong đó a_n là số phần tử có “kích thước” n trong lớp (tập hợp) $\mathcal{A}$

Ví dụ 37

(a) Hàm sinh của lớp các chuỗi nhị phân $\{0, 1\}^*$ là

$$G(x) = \sum_{n=0}^{\infty} 2^n x^n$$

- Có 2^n chuỗi nhị phân có độ dài n

(b) Hàm sinh của lớp giá trị các mặt của một xúc xắc thông

thường là $G(x) = \sum_{n=1}^6 x^n$

- Có 1 mặt giá trị 1, 1 mặt giá trị 2, 1 mặt giá trị 3, 1 mặt giá trị 4, 1 mặt giá trị 5, và 1 mặt giá trị 6

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

6

Hàm sinh

Một số lỗi thường gặp



Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

7 Hàm sinh

Một số lỗi thường gặp

Đếm bằng cách sử dụng hàm sinh

Các bài toán liên quan đến việc *đếm số cách chọn các phần tử trong một tập hợp* có thể được giải bằng cách sử dụng hàm sinh thông qua các lý luận để *đồng nhất số cách chọn n phần tử với hệ số của x^n trong hàm sinh*

- Hàm sinh cho phép làm việc với các hàm thay vì đếm một cách trực tiếp
- Khi làm việc với các hàm, có thể ứng dụng các thao tác biến đổi đại số (algebraic manipulation)



Tổng hai hàm sinh

Giả sử $\mathcal{A}$ và $\mathcal{B}$ là các lớp đối tượng không giao nhau, và $\mathcal{C} = \mathcal{A} \uplus \mathcal{B}$ là hợp của chúng (ký hiệu $\uplus$ biểu thị hợp của hai tập hợp không giao nhau). Nếu $A(x) = \sum_{n=0}^{\infty} a_n x^n$ và $B(x) = \sum_{n=0}^{\infty} b_n x^n$ lần lượt là các hàm sinh của $\mathcal{A}$ và $\mathcal{B}$, thì hàm sinh của $\mathcal{C}$ là

$$C(x) = A(x) + B(x) = \sum_{n=0}^{\infty} (a_n + b_n) x^n$$

- Có a_n đối tượng trong lớp $\mathcal{A}$ có kích thước n và b_n đối tượng trong lớp $\mathcal{B}$ có kích thước n , do đó tổng số đối tượng trong lớp $\mathcal{C}$ có kích thước n là $a_n + b_n$



Ví dụ 38

- $\mathcal{A} = \{\{1\}, \{2\}, \{1, 3\}, \{1, 3, 5\}\}$, $\mathcal{B} = \{\{4\}, \{5\}, \{6\}, \{4, 5\}\}$
(Chú ý là $\mathcal{A} \cap \mathcal{B} = \emptyset$)
- Kích thước của các phần tử (tập hợp) trong $\mathcal{A}$ và $\mathcal{B}$ là số thành viên trong phần tử (tập hợp) đó
- Hàm sinh của $\mathcal{A}$ là $A(x) = 2x^1 + 1x^2 + 1x^3$ (có 2 phần tử có kích thước 1, 1 phần tử có kích thước 2, và 1 phần tử có kích thước 3)
- Hàm sinh của $\mathcal{B}$ là $B(x) = 3x^1 + 1x^2$ (có 3 phần tử có kích thước 1, và 1 phần tử có kích thước 2)
- Hàm sinh của $\mathcal{C} = \mathcal{A} \uplus \mathcal{B}$ là
$$C(x) = A(x) + B(x) = 5x^1 + 2x^2 + 1x^3$$



Tích hai hàm sinh

Giả sử $\mathcal{A}$ và $\mathcal{B}$ là các lớp đối tượng và $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ là tích Descartes của chúng. Nếu $A(x) = \sum_{n=0}^{\infty} a_n x^n$ và $B(x) = \sum_{n=0}^{\infty} b_n x^n$ lần lượt là các hàm sinh của $\mathcal{A}$ và $\mathcal{B}$, thì hàm sinh của $\mathcal{C}$ là $C(x) = A(x) \cdot B(x) = \sum_{n=0}^{\infty} c_n x^n$ trong đó $c_n = \sum_{k=0}^n a_k b_{n-k}$

- Với $0 \leq k \leq n$ bất kỳ, có a_k đối tượng trong lớp $\mathcal{A}$ có kích thước k và b_{n-k} đối tượng trong lớp $\mathcal{B}$ có kích thước $n - k$. Do đó tổng số đối tượng trong lớp $\mathcal{C}$ có kích thước n là

$$c_n = \sum_{k=0}^n a_k b_{n-k}. \text{ (Kích thước của một cặp } (a, b) \text{ là tổng kích thước của } a \text{ và } b)$$

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

10 Hàm sinh

Một số lỗi thường gặp



Ví dụ 39

- $\mathcal{A} = \mathcal{B} = \{1, 2, \dots, 6\}$: tập hợp giá trị các mặt của một xúc xắc thông thường
- Hàm sinh của cả $\mathcal{A}$ và $\mathcal{B}$ đều là
$$G(x) = 1x^1 + 1x^2 + 1x^3 + 1x^4 + 1x^5 + 1x^6$$
(có 1 mặt giá trị 1, 1 mặt giá trị 2, ..., 1 mặt giá trị 6)
- $\mathcal{C} = \mathcal{A} \times \mathcal{B}$ là tập hợp các cặp (x, y) trong đó $x \in \mathcal{A}$ và $y \in \mathcal{B}$. Kích thước của mỗi cặp (x, y) là $x + y$
- Hàm sinh của $\mathcal{C}$ là
$$G(x)^2 = (1x^1 + 1x^2 + 1x^3 + 1x^4 + 1x^5 + 1x^6)^2$$

Hàm sinh



Ví dụ 40

Có bao nhiêu cách chọn k phần tử từ một tập hợp gồm n phần tử?

- Ta muốn xây dựng một hàm sinh $G(x)$ sao cho **hệ số của x^k trong $G(x)$ là số cách chọn k phần tử từ một tập hợp gồm n phần tử**
 $S = \{p_1, \dots, p_n\}$
- Gọi $\mathcal{A}_i$ ($1 \leq i \leq n$) là tập hợp các tập con của $\{p_i\}$. Hàm sinh của $\mathcal{A}_i$ là $G_i(x) = 1 + x$
 - Có 1 tập con của $\{p_i\}$ có kích thước 0 (tập rỗng) $\Leftrightarrow$ Hệ số của x^0 là 1
 - Có 1 tập con của $\{p_i\}$ có kích thước 1 ($\{p_i\}$) $\Leftrightarrow$ Hệ số của x^1 là 1
 - Có 0 tập con của $\{p_i\}$ có kích thước $k \geq 2 \Leftrightarrow$ Hệ số của x^k với $k \geq 2$ là 0
- Xét $\mathcal{C} = \mathcal{A}_1 \times \dots \times \mathcal{A}_n$.
 - $\mathcal{C}$ là tập hợp các bộ n phần tử $(S_1, \dots, S_n)$ trong đó $S_i \subseteq \{p_i\}$ với $1 \leq i \leq n$. Kích thước của mỗi bộ n phần tử là $\sum_{i=1}^n |S_i|$
 - Hàm sinh của $\mathcal{C}$ là $G(x) = G_1(x) \cdot \dots \cdot G_n(x) = (1+x)^n$
 - Tồn tại một song ánh $f: \mathcal{C} \rightarrow \mathcal{P}(S)$ cho bởi $f(S_1, \dots, S_n) = S_1 \cup \dots \cup S_n$
- Do đó, số cách chọn k phần tử từ một tập hợp gồm n phần tử (= số tập con k phần tử của S = số phần tử thuộc $\mathcal{C}$ có kích thước k) là hệ số của x^k trong khai triển của $G(x) = (1+x)^n$, và bằng C_n^k (Định lý nhị thức)

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

12 Hàm sinh

Một số lỗi thường gặp



Ví dụ 41

Phương trình $x_1 + x_2 + x_3 = 11$ có bao nhiêu nghiệm nguyên thỏa mãn $x_1 \geq 0$, $x_2 \geq 0$, và $x_3 \geq 0$?

- Ta muốn xây dựng một hàm sinh $G(x)$ cho phương trình sao cho *hệ số của x^{11} trong $G(x)$ là số nghiệm thỏa mãn phương trình*
- Gọi $\mathcal{A}_i$ ($1 \leq i \leq 3$) là tập hợp các giá trị có thể của x_i . Hàm sinh của $\mathcal{A}_i$ là $G_i(x) = 1 + x + \cdots + x^{11}$
- Xét $\mathcal{C} = \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3$
 - $\mathcal{C}$ là tập hợp các bộ ba (x_1, x_2, x_3) trong đó $x_i \in \mathcal{A}_i$ với $1 \leq i \leq 3$. Kích thước của mỗi bộ ba là $x_1 + x_2 + x_3$
 - Hàm sinh của $\mathcal{C}$ là $G(x) = G_1(x) \cdot G_2(x) \cdot G_3(x) = (1 + x + \cdots + x^{11})^3$
 - Mỗi nghiệm của phương trình ứng với một bộ có kích thước 11 trong $\mathcal{C}$

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

13 Hàm sinh

Một số lỗi thường gặp



Ví dụ 41 (tiếp)

Phương trình $x_1 + x_2 + x_3 = 11$ có bao nhiêu nghiệm nguyên thỏa mãn $x_1 \geq 0$, $x_2 \geq 0$, và $x_3 \geq 0$?

- Số nghiệm của phương trình là hệ số của x^{11} trong khai triển của $G(x) = (1 + x + \cdots + x^{11})^3$
- Ta có

$$\begin{aligned} G(x) &= (1 + x + \cdots + x^{11})^3 \\ &= \left(\frac{1 - x^{12}}{1 - x} \right)^3 \\ &= (1 - x^{12})^3 (1 - x)^{-3} \end{aligned}$$

Để tìm hệ số của x^{11} trong khai triển của $G(x)$ ở Ví dụ 41, ta cần một số định nghĩa và định lý

Hàm sinh



Hệ số nhị thức mở rộng

Với $n \in \mathbb{R}$ bất kỳ và $k \geq 0$, **hệ số nhị thức tổng quát (generalized binomial coefficient)** C_n^k được định nghĩa như sau

$$C_n^k = \frac{n(n-1)\dots(n-k+1)}{k!}$$

■ Ví dụ, $C_{-2}^5 = \frac{(-2)(-3)(-4)(-5)(-6)}{5!} = -6$

Mệnh đề 15

Nếu n là một số nguyên dương thì

$$C_{-n}^k = (-1)^k C_{n+k-1}^k$$

Định lý 16: Định lý nhị thức tổng quát

Với mọi $n \in \mathbb{R}$,

$$(1+x)^n = \sum_{k=0}^{\infty} C_n^k x^k$$

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

15 Hàm sinh

Một số lỗi thường gặp

Hàm sinh



Ví dụ 42

Tìm hệ số của x^{11} trong khai triển của $(1 - x)^{-3}$

- Hệ số của x^r trong khai triển của $(1 - x)^{-3}$ là

$$(-1)^r C_{-3}^r = (-1)^r [(-1)^r \cdot C_{3+r-1}^r] = \frac{(r+2)(r+1)}{2}$$

- Suy ra, hệ số của x^{11} trong khai triển của $(1 - x)^{-3}$ là $(11+2)(11+1)/2 = 78$

Ví dụ 43 (Tiếp tục Ví dụ 41)

Tìm hệ số của x^{11} trong khai triển của

$$G(x) = (1 - x^{12})^3 (1 - x)^{-3}$$

- Ta có $(1 - x^{12})^3 = 1 - 3x^{12} + 3x^{24} - x^{36}$
- Cách duy nhất để thu được x^{11} là lấy x^0 trong khai triển của $(1 - x^{12})^3$ nhân với x^{11} trong khai triển của $(1 - x)^{-3}$
- Do đó, hệ số của x^{11} trong khai triển của $G(x)$ là hệ số của x^{11} trong khai triển của $(1 - x)^{-3}$, và bằng 78 (Ví dụ 42)

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

16 Hàm sinh

Một số lỗi thường gặp

Hàm sinh



Ví dụ 44

Phương trình $x_1 + x_2 + x_3 = 6$ có bao nhiêu nghiệm nguyên thỏa mãn $-1 \leq x_1 \leq 2$ và $1 \leq x_2, x_3 \leq 4$?

■ Hàm sinh

$$G(x) = (x^{-1} + x^0 + x^1 + x^2)(x^1 + x^2 + x^3 + x^4)^2$$

■ Số nghiệm nguyên của phương trình thỏa mãn điều kiện đề ra là hệ số của x^6 trong khai triển của $G(x)$

■ Ta có

$$\begin{aligned} G(x) &= (x^{-1} + x^0 + x^1 + x^2)(x^1 + x^2 + x^3 + x^4)^2 \\ &= \frac{1}{x^2}(x^1 + x^2 + x^3 + x^4)^3 \\ &= \frac{x^3}{x^2}(1 + x + x^2 + x^3)^3 \\ &= x(1 + x + x^2 + x^3)^3 \end{aligned}$$

Bài tập 46

Hoàn thành Ví dụ 44 bằng cách tìm hệ số của x^5 trong khai triển của $(1 + x + x^2 + x^3)^3$

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

17

Hàm sinh

Một số lỗi thường gặp

Hàm sinh



Ví dụ 45

Có bao nhiêu cách để bỏ n quả vào một túi thỏa mãn các ràng buộc sau?

- Các quả này chỉ thuộc bốn loại: táo, chuối, cam, lê
- Số lượng táo phải là số chẵn
- Số lượng chuối phải là bội số của 5
- Có nhiều nhất bốn quả cam
- Có nhiều nhất một quả lê
- Ví dụ, có 7 cách thực hiện với 6 quả:

Táo	6	4	4	2	2	0	0
Chuối	0	0	0	0	0	5	5
Cam	0	2	1	4	3	1	0
Lê	0	0	1	0	1	0	1

- Có *quá nhiều điều kiện*. Có vẻ như rất khó nếu đếm theo cách thông thường

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

18 Hàm sinh

Một số lỗi thường gặp

Hàm sinh



Ví dụ 45 (tiếp)

■ Hàm sinh cho việc lựa chọn các loại quả

■ Táo: $G_T(x) = 1 + x^2 + x^4 + \dots = \frac{1}{1 - x^2}$

■ Chuối: $G_C(x) = 1 + x^5 + x^{10} + \dots = \frac{1}{1 - x^5}$

■ Cam: $G_M(x) = 1 + x + x^2 + x^3 + x^4 = \frac{1 - x^5}{1 - x}$

■ Lê: $G_L(x) = 1 + x = \frac{1 - x^2}{1 - x}$

■ Hàm sinh cho tổng số quả

$$\begin{aligned} G(x) &= G_T(x) \cdot G_C(x) \cdot G_M(x) \cdot G_L(x) \\ &= \frac{1}{1 - x^2} \cdot \frac{1}{1 - x^5} \cdot \frac{1 - x^5}{1 - x} \cdot \frac{1 - x^2}{1 - x} = \frac{1}{(1 - x)^2} \end{aligned}$$

Bài tập 47

Hoàn thành Ví dụ 45 bằng cách tìm hệ số của x^n trong khai triển của $G(x) = \frac{1}{(1 - x)^2}$

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

19 Hàm sinh

Một số lỗi thường gặp

Hàm sinh



Bài tập 48

Tìm hệ số

(a) của x^4 trong khai triển của $(1 - x)^{-2}$

(b) của x^n trong khai triển của $(1 + x)^{-4}$

(c) của x^n trong khai triển của $\frac{1+x}{(1-2x)^5}$

(Gợi ý: $(1+x)/((1-2x)^5) = (1-2x)^{-5} + x(1-2x)^{-5}$. Tìm hệ số của x^n trong khai triển của từng số hạng và cộng các kết quả tìm được)

Bài tập 49

Sử dụng hàm sinh để đếm số nghiệm nguyên của $x_1 + x_2 + x_3 = 11$ thỏa mãn $0 \leq x_1 \leq 2$, $x_2 \geq 0$, và $x_3 \geq 0$

Bài tập 50

Xác định số cách để có được tổng 12 khi tung 3 viên xúc xắc bình thường bằng cách sử dụng hàm sinh

Bài tập 51 (★)

Sử dụng hàm sinh để đếm số nghiệm nguyên của $x_1 + x_2 + x_3 = 11$ thỏa mãn $x_1 \geq -2$, $x_2 \geq 1$, và $x_3 \geq 0$

Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

20 Hàm sinh

Một số lỗi thường gặp

Một số lỗi thường gặp



Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Hàm sinh

21

Một số lỗi thường gặp

Chú ý

Tham khảo từ tài liệu “Common Mistakes in Discrete Mathematics” (https://highereducation.com/sites/default/files/125967651x/1106131/Common_Mistakes_in_Discrete_Math.pdf)

(a) Vẽ sơ đồ không chính xác khi giải quyết các bài toán đếm

- Sơ đồ rất hữu ích trong toàn bộ toán học, và việc vẽ sơ đồ hầu như luôn là cách tốt để bắt đầu giải quyết một bài toán; do đó việc không vẽ sơ đồ cũng có thể được coi là một lỗi phổ biến.
- Ví dụ, bạn nên vẽ một hàng gồm sáu ô trống (không phải năm) làm khuôn mẫu để xây dựng các từ có độ dài 6 mà các ký hiệu được chọn từ một tập hợp gồm năm phần tử. Sơ đồ cây đôi khi cũng rất hữu ích

Một số lỗi thường gặp (tiếp)



Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Hàm sinh

22

Một số lỗi thường gặp

(b) Không xác định liệu thứ tự có quan trọng hay không trong việc giải quyết bài toán đếm

- Ví dụ, nếu chúng ta được hỏi số cách để viết 7 dưới dạng tổng các số nguyên dương, thì chúng ta cần biết liệu $3 + 2 + 2$ và $2 + 3 + 2$ có được coi là cùng một cách hay là các cách khác nhau
- Hãy đọc kỹ đề bài để hiểu rõ những gì đang được đếm. Giải quyết mọi điểm không rõ ràng bằng cách nêu rõ các giả định dường như thiếu từ đề bài

(c) Không xác định liệu có cho phép lặp lại hay không trong việc giải quyết bài toán đếm

- Ví dụ, nếu chúng ta được hỏi số cách để chọn năm chiếc bánh donut từ một cửa hàng bán tám loại bánh donut khác nhau, chúng ta cần biết liệu có được phép chọn nhiều hơn một bánh donut cùng loại hay không

Một số lỗi thường gặp (tiếp)



Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Hàm sinh

23

Một số lỗi thường gặp

- Hãy đọc kỹ đề bài để hiểu rõ những gì đang được đếm. Giải quyết mọi điểm không rõ ràng bằng cách nêu rõ các giả định dường như thiếu từ đề bài

(d) Đếm mỗi phần tử trong tập hợp đang xét nhiều hơn một lần, không nhận ra rằng cần phải điều chỉnh cho việc đếm hai lần

- Ví dụ, nếu chúng ta đếm số cái bắt tay theo từng người, thì chúng ta cần nhận ra rằng mỗi cái bắt tay đã được đếm hai lần, mỗi lần cho một người tham gia

(e) Đếm mỗi phần tử trong tập hợp đang xét nhiều hơn một lần, không nhận ra rằng cần sử dụng nguyên tắc bù-trừ (inclusion-exclusion principle)

Một số lỗi thường gặp (tiếp)



Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và

Định lý nhị thức

Nguyên lý bù trừ tổng quát

Hàm sinh

24

Một số lỗi thường gặp

- Ví dụ, nếu chúng ta được biết có 26 sinh viên chuyên ngành khoa học máy tính và 34 sinh viên chuyên ngành toán học tại một trường đại học, thì có thể không có $26 + 34 = 60$ người học chuyên ngành khoa học máy tính hoặc toán học, vì hai con số này có thể bao gồm cả những người học đồng thời cả hai ngành
- Để tính chính xác tổng số người học các chuyên ngành này, chúng ta cần trừ đi số lượng người học cả hai ngành từ tổng này

(f) Sử dụng nguyên lý chuồng bồ câu hoặc nguyên lý chuồng bồ câu tổng quát không chính xác

- Nên xác định rõ đâu là “chim bồ câu” và đâu là “chuồng”
- Ví dụ, để tìm số quân bài tối thiểu phải chọn để đảm bảo có ít nhất sáu quân bài cùng chất, các “chuồng” là các chất bài và các quân bài là các “chim bồ câu” (câu trả lời là 21).

(g) Lỗi khi áp dụng nguyên lý bù-trừ

Một số lỗi thường gặp (tiếp)



Các phương pháp đếm

Hoàng Anh Đức

Tam giác Pascal và
Định lý nhị thức

Nguyên lý bù trừ tổng quát

Hàm sinh

25

Một số lỗi thường gặp

- Việc tin rằng $|A \cup B| = |A| + |B|$ luôn đúng liên quan đến suy nghĩ mong muốn rằng mọi phép toán đều phân phối (hoặc có hành vi đơn giản, dễ chịu nào đó) đối với mọi phép toán khác. Đẳng thức này chỉ đúng khi A và B rời nhau
- Nhầm lẫn dấu của các số hạng khi áp dụng nguyên lý bù-trừ. Lưu ý rằng các dấu thay đổi xen kẽ khi chúng ta xét hợp của càng nhiều tập hợp
- Không đưa đầy đủ các số hạng khi áp dụng nguyên lý bù-trừ. Nếu có n tập hợp được xét, thì có gần 2^n số hạng khác nhau trong phương trình

1. Logic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Đệ quy
4. Thuật toán I: Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp
5. Thuật toán II: Thuật toán tham lam, thuật toán đệ quy
6. Lý thuyết số cơ bản
7. Các phương pháp đếm
8. Lý thuyết đồ thị I: Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông
9. Lý thuyết đồ thị II: Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị
10. Lý thuyết đồ thị III: Cây
11. Đại số Boole

VNU-HUS MAT3500: Toán rời rạc

Lý thuyết đồ thị I

Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn



Nội dung



Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa đồ thị và một số ví dụ



Lý thuyết đồ thị I

Hoàng Anh Đức

- Một **đồ thị (graph)** G bao gồm một tập các **đỉnh (vertex)** hoặc **nút (node)** V và một tập các **cạnh (edge)** E nối các (cặp) đỉnh với nhau
 - Ký hiệu $G = (V, E)$
 - Các ký hiệu $V(G)$ và $E(G)$ cũng được sử dụng để chỉ các tập đỉnh và cạnh của một đồ thị G
- Biểu diễn hình học của đồ thị:
 - Các đỉnh được biểu diễn bằng các vòng tròn nhỏ
 - Mỗi cạnh vô hướng được biểu diễn bằng một đoạn thẳng nối hai đỉnh
 - Mỗi cạnh có hướng được biểu diễn bằng một mũi tên nối hai đỉnh
- Có nhiều thuật ngữ khác nhau và thường không thống nhất

2

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa đồ thị và một số ví dụ



Lý thuyết đồ thị I

Hoàng Anh Đức

- G được gọi là **đồ thị hữu hạn (finite graph)** nếu V là tập hữu hạn và là **đồ thị vô hạn (infinite graph)** nếu V là tập vô hạn. Chúng ta chỉ đề cập đến các đồ thị hữu hạn
- $G = (\emptyset, \emptyset)$ được gọi là **đồ thị rỗng (null graph)**
 - Trong nhiều trường hợp, việc coi $G = (\emptyset, \emptyset)$ là đồ thị hay không phụ thuộc vào ngữ cảnh [Harary and Read 1973]
- Trong nhiều trường hợp, thuật ngữ **đồ thị rỗng (empty graph hoặc edgeless graph)** được sử dụng để chỉ đồ thị $G = (V, \emptyset)$ có ít nhất một đỉnh ($|V| \geq 1$) nhưng không có cạnh
- Thông thường, trừ trường hợp đặc biệt, ta luôn giả thiết **một đồ thị có ít nhất một đỉnh**

Giới thiệu

3 Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa đồ thị và một số ví dụ



Lý thuyết đồ thị I

Hoàng Anh Đức

Với một tập V , gọi $[V]^k$ là *tập hợp tất cả các tập con k phần tử của V* . (Nói cách khác, $[V]^k$ là tập hợp tất cả các tổ hợp chập k của V)

Đồ thị vô hướng

Một *đồ thị vô hướng (undirected graph)* $G = (V, E)$ bao gồm một tập khác rỗng V gồm các *đỉnh (vertex)* (hoặc *nút (node)*), và một tập $E \subseteq [V]^2$ gồm các *cạnh vô hướng (undirected edge)*.

Mỗi cạnh $e = uv \in E$ (hoặc $e = \{u, v\} \in E$) có hai đỉnh phân biệt $u \neq v$ là các *đầu nút (endpoint)* của e . Ta nói các đỉnh u, v là *liên kề (adjacent)* trong đồ thị G , và cạnh e gọi là cạnh *liên thuộc (incident)* với các đỉnh u, v

Giới thiệu

4 Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm
Đồ thị mới từ đồ thị cũ
Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu
Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề
Ma trận kề
Ma trận liên thuộc
Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi
Liên thông trong đồ thị vô hướng
Liên thông trong đồ thị có hướng
Đường đi và sự đẳng cấu
Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa đồ thị và một số ví dụ



Lý thuyết đồ thị I

Hoàng Anh Đức

Đồ thị có hướng

Một **đồ thị có hướng** (*directed graph* hoặc *digraph*) $G = (V, E)$ bao gồm một tập khác rỗng V gồm các **đỉnh** (*vertex*) (hoặc **nút** (*node*)) và một tập $E \subseteq V \times V$ gồm các **cạnh có hướng** (*directed edge*) (hoặc **cung** (*arc*)).

Mỗi cạnh có hướng $(u, v) \in E$ có một **đỉnh đầu** (*start vertex* hoặc **tail vertex**) u và **một đỉnh cuối** (*end vertex* hoặc **head vertex**) v

5

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

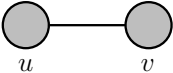
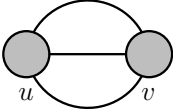
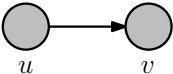
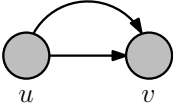
Giới thiệu

Định nghĩa đồ thị và một số ví dụ



Lý thuyết đồ thị I

Hoàng Anh Đức

	Cạnh	Cạnh song song	Khuyên
Vô hướng			
Có hướng			

Hình: Một số loại cạnh trong đồ thị

Giới thiệu

6 Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm
Đồ thị mới từ đồ thị cũ
Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu
Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề
Ma trận kề
Ma trận liên thuộc
Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi
Liên thông trong đồ thị vô hướng
Liên thông trong đồ thị có hướng
Đường đi và sự đẳng cấu
Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa đồ thị và một số ví dụ



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

7 Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm
Đồ thị mới từ đồ thị cũ
Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu
Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề
Ma trận kề
Ma trận liên thuộc
Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi
Liên thông trong đồ thị vô hướng
Liên thông trong đồ thị có hướng
Đường đi và sự đẳng cấu
Đếm số đường đi giữa các đỉnh

	Loại	Cạnh	Có cạnh song song?	Có khuyên?
1	Đơn đồ thị vô hướng	Vô hướng	Không	Không
2	Đa đồ thị vô hướng	Vô hướng	Có	Không
3	Đa đồ thị vô hướng có khuyên	Vô hướng	Có	Có
4	Đơn đồ thị có hướng	Có hướng	Không	Có
5	Đơn đồ thị có hướng	Có hướng	Không	Không
6	Đa đồ thị có hướng	Có hướng	Có	<i>Không¹</i>
7	Đa đồ thị có hướng và có khuyên	Có hướng	Có	Có
8	Đồ thị hỗn hợp	Cả hai	Có	Có

- Định nghĩa đa đồ thị có hướng khác với định nghĩa trong [Rosen 2012]
- Chủ yếu quan tâm đến các đồ thị sau:
 - đơn đồ thị vô hướng (simple, undirected graph)
 - đồ thị có hướng (directed graph hoặc digraph)

¹ Khác với phân loại trong [Rosen 2012]

Giới thiệu

Định nghĩa đồ thị và một số ví dụ



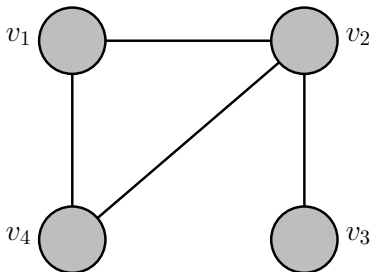
Lý thuyết đồ thị I

Hoàng Anh Đức

Ví dụ 1 (Đơn đồ thị vô hướng (simple undirected graph))

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_4, v_2v_3, v_2v_4\}$$



Hình: Chỉ có các cạnh **vô hướng**; có **nhiều nhất một cạnh** nối hai đỉnh phân biệt bất kỳ; và không có **khuyên (loop)**

Giới thiệu

8 Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

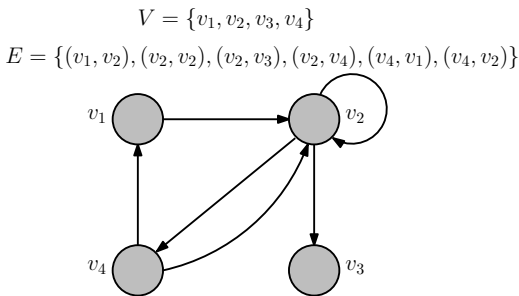
Định nghĩa đồ thị và một số ví dụ



Lý thuyết đồ thị I

Hoàng Anh Đức

Ví dụ 2 (Đồ thị có hướng (và có khuyên) (directed graph (with loops)))



Hình: Chỉ có các cạnh **có hướng**; có **nhều nhất một cạnh có hướng** nối từ một đỉnh bất kỳ sang một đỉnh khác bất kỳ; và **có khuyên**

Giới thiệu

9 Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm
Đồ thị mới từ đồ thị cũ
Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu
Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề
Ma trận kề
Ma trận liên thuộc
Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi
Liên thông trong đồ thị vô hướng
Liên thông trong đồ thị có hướng
Đường đi và sự đẳng cấu
Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

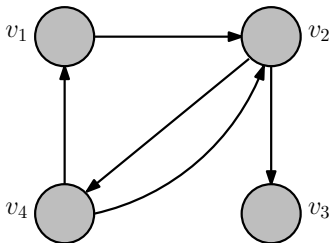


Lý thuyết đồ thị I

Hoàng Anh Đức

Ví dụ 3 (Đơn đồ thị có hướng (simple directed graph))

$$V = \{v_1, v_2, v_3, v_4\}$$
$$E = \{(v_1, v_2), (v_2, v_3), (v_2, v_4), (v_4, v_1), (v_4, v_2)\}$$



Hình: Chỉ có các cạnh *có hướng*; có *nhều nhất một cạnh có hướng* nối từ một đỉnh bất kỳ sang một đỉnh khác bất kỳ; và *không có khuyên*

Giới thiệu

10 Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Mã trận kề

Mã trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa đồ thị và một số ví dụ



Lý thuyết đồ thị I

Hoàng Anh Đức

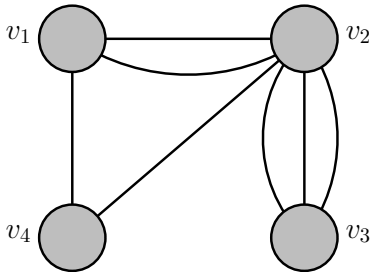
Ví dụ 4 (Đa đồ thị vô hướng (undirected multigraph))

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_4, v_2v_3, v_2v_4\}$$

$$m(v_1v_2) = 2, m(v_2v_3) = 3$$

$$m(v_1v_4) = m(v_2v_4) = 1$$



Hình: Chỉ có các cạnh **vô hướng**; có thể có **nhiều cạnh** nối giữa hai đỉnh bất kỳ; và **không có khuyên**

Giới thiệu

11 Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm
Đồ thị mới từ đồ thị cũ
Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu
Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề
Ma trận kề
Ma trận liên thuộc
Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi
Liên thông trong đồ thị vô hướng
Liên thông trong đồ thị có hướng
Đường đi và sự đẳng cấu
Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

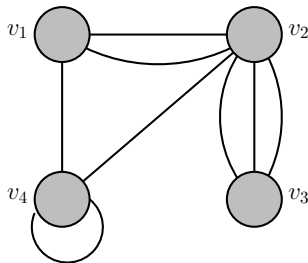


Lý thuyết đồ thị I

Hoàng Anh Đức

Ví dụ 5 (Đa đồ thị vô hướng có khuyên (undirected pseudograph))

$$\begin{aligned}V &= \{v_1, v_2, v_3, v_4\} \\E &= \{v_1v_2, v_1v_4, v_2v_3, v_2v_4, v_4v_4\} \\m(v_1v_2) &= 2, m(v_2v_3) = 3 \\m(v_1v_4) &= m(v_2v_4) = m(v_4, v_4) = 1\end{aligned}$$



Hình: Chỉ có các cạnh **vô hướng**; có thể có **nhiều cạnh** nối giữa hai đỉnh bất kỳ; và **có khuyên** (có thể có nhiều khuyên tại một đỉnh)

Giới thiệu

12 Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa đồ thị và một số ví dụ



Lý thuyết đồ thị I

Hoàng Anh Đức

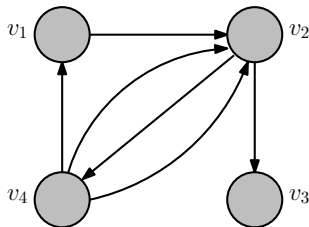
Ví dụ 6 (Đa đồ thị có hướng (directed multigraph))

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{(v_1, v_2), (v_2, v_3), (v_2, v_4), (v_4, v_1), (v_4, v_2)\}$$

$$m(v_1, v_2) = m(v_2, v_3) = m(v_2, v_4) = m(v_4, v_1) = 1$$

$$m(v_4, v_2) = 2$$



Hình: Chỉ có các cạnh **có hướng**; có thể có **nhiều cạnh** nối giữa hai đỉnh bất kỳ; và **không có khuyên** (khác với định nghĩa trong sách của Rosen)

Giới thiệu

13

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa đồ thị và một số ví dụ



Lý thuyết đồ thị I

Hoàng Anh Đức

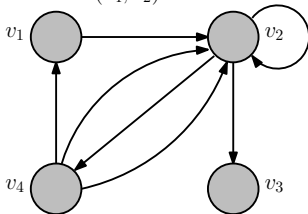
Ví dụ 7 (Đa đồ thị có hướng và có khuyên (directed pseudograph))

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{(v_1, v_2), (v_2, v_2), (v_2, v_3), (v_2, v_4), (v_4, v_1), (v_4, v_2)\}$$

$$m(v_1, v_2) = m(v_2, v_2) = m(v_2, v_3) = m(v_2, v_4) = m(v_4, v_1) = 1$$

$$m(v_4, v_2) = 2$$



Hình: Chỉ có các cạnh **có hướng**; có thể có **nhiều cạnh** nối giữa hai đỉnh bất kỳ; và **có khuyên** (có thể có nhiều khuyên tại một đỉnh)

Giới thiệu

14 Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa và khái niệm



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

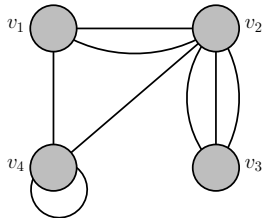
Đếm số đường đi giữa các đỉnh

Cho $G = (V, E)$ là một đồ thị vô hướng

- Tập hợp các đỉnh kề với đỉnh v của G , ký hiệu $N(v)$, được gọi là **tập láng giềng (neighborhood)** của v .
- Với một tập các đỉnh $A \subseteq V$, ta ký hiệu $N(A)$ để chỉ tập các đỉnh liên kề với ít nhất một đỉnh trong A . Nói cách khác, $N(A) = \bigcup_{v \in A} N(v)$
- **Bậc (degree)** của một đỉnh v , ký hiệu $\deg(v)$, là số cạnh của G liên thuộc với đỉnh đó. Một khuyên tại đỉnh v (một cạnh nối v với chính nó) đóng góp 2 vào bậc của v

Ví dụ 8

- $N(v_1) = \{v_2, v_4\}$,
 $N(v_2) = \{v_1, v_3, v_4\}$,
 $N(v_3) = \{v_2\}$,
 $N(v_4) = \{v_1, v_2, v_4\}$
- $\deg(v_1) = \deg(v_3) = 3$,
 $\deg(v_2) = 6, \deg(v_4) = 4$



Giới thiệu

Định nghĩa và khái niệm



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

16

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

- Một đỉnh bậc 0 được gọi là một **đỉnh cô lập (isolated vertex)**
- Một đỉnh bậc 1 được gọi là một **đỉnh treo (pendant vertex)**
- Có thể thêm tên đồ thị vào các ký hiệu để nhấn mạnh đồ thị đang xét: ví dụ, $\deg_G(v)$ là bậc của đỉnh v trong đồ thị G , $N_G(v)$ là tập láng giềng của v trong đồ thị G , và $N_G(A)$ là tập láng giềng của A trong đồ thị G , v.v.

Chú ý

Thường có sự nhầm lẫn rằng $\deg(v) = |N(v)|$ với mọi đỉnh v trong một đồ thị vô hướng G . Chú ý rằng điều này không đúng trong trường hợp G có khuyên hoặc cạnh song song. Nói cách khác, $\deg(v) = |N(v)|$ với mọi đỉnh v trong G khi G là đơn đồ thị vô hướng



Định lý 1: Định lý bắt tay (Handshaking Lemma)

Cho $G = (V, E)$ là một đồ thị vô hướng. Ta có

$$2|E| = \sum_{v \in V} \deg(v)$$

Chứng minh.

- Với mỗi cạnh $e = uv \in E$, e được đếm chính xác hai lần trong $\sum_{v \in V} \deg(v)$: một lần trong $\deg(u)$ và một lần trong $\deg(v)$
- Do đó, cả hai vế của đẳng thức trên đều bằng hai lần số cạnh của G

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Mã trận kề

Mã trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa và khái niệm



Định lý 2

Một đồ thị vô hướng có một số chẵn các đỉnh có bậc lẻ

Chứng minh.

- Gọi V_1 là tập các đỉnh bậc chẵn và V_2 là tập các đỉnh bậc lẻ trong đồ thị vô hướng $G = (V, E)$
- Ta có $2|E| = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v)$
- $\sum_{v \in V_1} \deg(v)$ là một số chẵn, vì V_1 là tập tất cả các đỉnh có bậc chẵn
- Do đó, $\sum_{v \in V_2} \deg(v)$ là một số chẵn, do $2m$ và $\sum_{v \in V_1} \deg(v)$ đều là số chẵn
- Do V_2 là tập các đỉnh bậc lẻ, để $\sum_{v \in V_2} \deg(v)$ chẵn, cần phải có một số chẵn các đỉnh bậc lẻ

Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

18

78

Giới thiệu

Định nghĩa và khái niệm



Ví dụ 9

Có bao nhiêu cạnh trong một đồ thị vô hướng có 10 đỉnh, mỗi đỉnh có bậc 6?

Ví dụ 10

Nếu một đồ thị vô hướng có 5 đỉnh thì liệu mỗi đỉnh có thể có bậc 3 hay không?

Bài tập 1

Cho G là một đồ thị vô hướng có n đỉnh và m cạnh. Gọi $\Delta(G)$ và $\delta(G)$ lần lượt là bậc lớn nhất và nhỏ nhất của một đỉnh của G . Chứng minh rằng $\delta(G) \leq 2m/n \leq \Delta(G)$

Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

19

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Giới thiệu

Định nghĩa và khái niệm



Lý thuyết đồ thị I

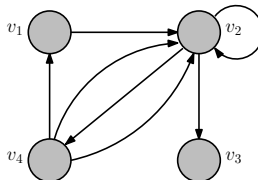
Hoàng Anh Đức

Cho $G = (V, E)$ là một đồ thị có hướng

- **Bậc vào (in-degree)** của một đỉnh v , ký hiệu $\deg^-(v)$ là số các cạnh có đỉnh cuối (tail vertex) là v
- **Bậc ra (out-degree)** của một đỉnh v , ký hiệu $\deg^+(v)$ là số các cạnh có đỉnh đầu (head vertex) là v
- Một khuyên ở đỉnh v đóng góp 1 vào bậc vào và 1 vào bậc ra của v

Ví dụ 11

- $\deg^-(v_1) = \deg^-(v_3) =$
 $\deg^-(v_4) = 1,$
 $\deg^-(v_2) = 4$
- $\deg^+(v_1) = 1,$
 $\deg^+(v_2) = \deg^+(v_4) = 3,$
 $\deg^+(v_3) = 0$



Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

Định nghĩa và khái niệm



Lý thuyết đồ thị I

Hoàng Anh Đức

Định lý 3

Cho $G = (V, E)$ là một đồ thị có hướng. Ta có

$$|E| = \sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v)$$

Chứng minh.

- Mỗi cạnh có hướng $e = (u, v) \in E$ đóng góp 1 vào $\deg^-(v)$ và 1 vào $\deg^+(u)$, với $u, v \in V$
- Do đó, $|E| = \text{tổng các bậc vào} = \text{tổng các bậc ra}$

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

Đồ thị mới từ đồ thị cũ



Lý thuyết đồ thị I

Hoàng Anh Đức

Cho đơn đồ thị vô hướng $G = (V, E)$

- $H = (W, F)$ là một **đồ thị con (subgraph)** của G (hay G **chứa** H), ký hiệu $H \subseteq G$, nếu $W \subseteq V$ và $F \subseteq E$. Nếu $W = V$ thì H được gọi là **đồ thị con bao trùm (spanning subgraph)** của G
- $H = (W, F)$ là một **đồ thị con thực sự (proper subgraph)** của G , ký hiệu $H \subset G$, nếu $H \subseteq G$, và $W \subset V$ hoặc $F \subset E$
- $H = (W, F)$ là một **đồ thị con cảm sinh (induced subgraph)** của G nếu $H \subseteq G$ và với mọi cặp đỉnh $u, v \in W$, $uv \in F$ khi và chỉ khi $uv \in E$. Ta cũng nói **H là đồ thị con của G cảm sinh bởi W** và viết $H = G[W]$

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Mã trận kề

Mã trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

22

78

Giới thiệu

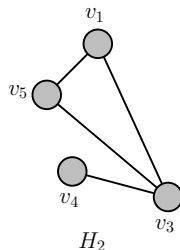
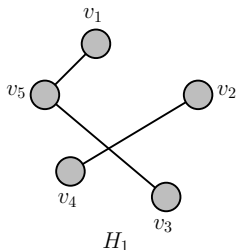
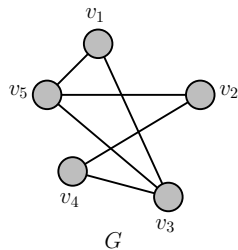
Đồ thị mới từ đồ thị cũ



Lý thuyết đồ thị I

Hoàng Anh Đức

Ví dụ 12



Hình: H_1 là đồ thị con thực sự và là đồ thị con bao trùm của G nhưng không phải đồ thị con cảm sinh. H_2 là đồ thị con thực sự và là đồ thị con cảm sinh của G nhưng không phải đồ thị con bao trùm

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

23 Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

Đồ thị mới từ đồ thị cũ

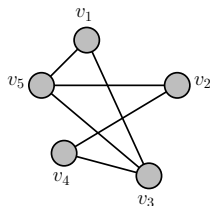


Lý thuyết đồ thị I

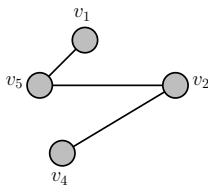
Hoàng Anh Đức

Cho đơn đồ thị $G = (V, E)$ vô hướng và các tập $V' \subseteq V$ và $E' \subseteq E$

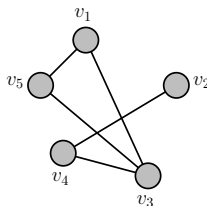
- Đồ thị $G - V'$ là đồ thị thu được bằng cách *xóa các đỉnh trong V' và các cạnh liên thuộc với chúng*. Với một đỉnh $v \in V'$, ta viết $G - v$ thay vì $G - \{v\}$
- Đồ thị $G - E'$ là đồ thị thu được bằng cách *xóa các cạnh trong E'* . Với một cạnh $e \in E'$, ta viết $G - e$ thay vì $G - \{e\}$



G



$G - v_3$



$G - v_2v_5$

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

24

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Giới thiệu

Đồ thị mới từ đồ thị cũ

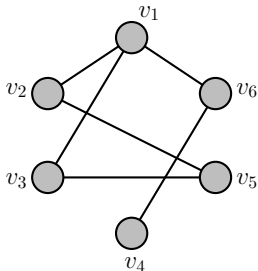


Lý thuyết đồ thị I

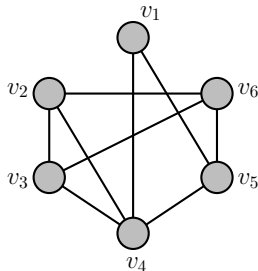
Hoàng Anh Đức

Cho đơn đồ thị vô hướng $G = (V, E)$

- **Đồ thị bù (complement graph)** của G , ký hiệu $\overline{G} = (\overline{V}, \overline{E})$, là đồ thị có tập đỉnh $\overline{V} = V$ và tập cạnh $\overline{E} = [V]^2 \setminus E = \{uv \mid u, v \in V \text{ và } uv \notin E\}$
- $\overline{G}$ là đồ thị thu được từ G bằng cách **xóa các cạnh trong E** và **thêm các cạnh trong $\overline{E} = [V]^2 \setminus E$**



G



$\overline{G}$

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

25

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Giới thiệu

Đồ thị mới từ đồ thị cũ

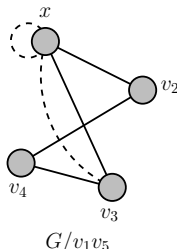
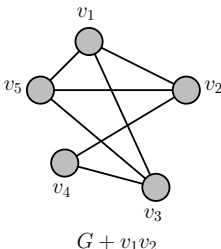
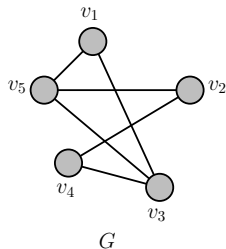


Lý thuyết đồ thị I

Hoàng Anh Đức

Cho đơn đồ thị $G = (V, E)$ vô hướng và tập $E' \subseteq [V]^2 \setminus E$

- Đồ thị $G + E'$ là đồ thị thu được bằng cách **thêm các cạnh trong E'** . Với $f \in E'$, ta viết $G + f$ thay vì $G + \{f\}$
- Đồ thị G/e là đồ thị thu được bằng **phép co (contraction) cạnh $e = uv \in E$**
 - gộp hai đỉnh u, v thành một đỉnh mới x , các cạnh kề với u và kề với v chuyển thành cạnh kề với x
 - xóa các khuyên tạo thành sau phép gộp
 - giữ lại một cạnh duy nhất trong số các cạnh song song



Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

26

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Giới thiệu

Một số đơn đồ thị đặc biệt



Lý thuyết đồ thị I

Hoàng Anh Đức

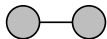
Đồ thị đầy đủ

Đồ thị đầy đủ (complete graph) n đỉnh, ký hiệu K_n , là một đơn đồ thị chứa đúng một cạnh nối mỗi cặp đỉnh phân biệt.

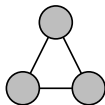
Cụ thể, $V(K_n) = \{v_1, \dots, v_n\}$ và $E(K_n) = \{v_i v_j \mid 1 \leq i < j \leq n\}$



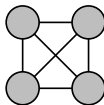
K_1



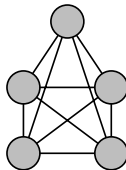
K_2



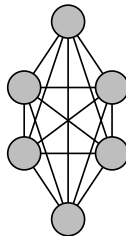
K_3



K_4



K_5



K_6

27

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Giới thiệu

Một số đơn đồ thị đặc biệt

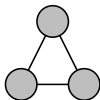


Lý thuyết đồ thị I

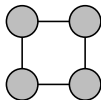
Hoàng Anh Đức

Chu trình

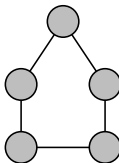
Một **chu trình (cycle)** n đỉnh với $n \geq 3$, ký hiệu C_n , là một đồ thị với tập đỉnh $V(C_n) = \{v_1, v_2, \dots, v_n\}$ và tập cạnh $E(C_n) = \{v_i v_{i+1} \mid 1 \leq i \leq n-1\} \cup \{v_n v_1\}$



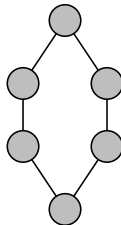
C_3



C_4



C_5



C_6

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

28

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Giới thiệu

Một số đơn đồ thị đặc biệt

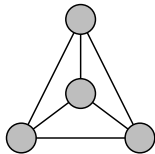


Lý thuyết đồ thị I

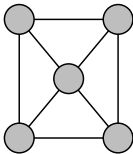
Hoàng Anh Đức

Đồ thị bánh xe

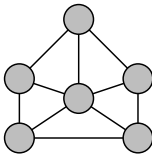
Một **đồ thị bánh xe (wheel)** gồm $n + 1$ đỉnh với $n \geq 3$, ký hiệu W_n , là một đồ thị thu được bằng cách thêm một đỉnh mới vào C_n và nối đỉnh đó với mọi đỉnh của C_n bằng các cạnh mới. Cụ thể, $V(W_n) = \{v_1, \dots, v_n\} \cup \{w\}$ ($w \notin \{v_1, \dots, v_n\}$) và $E(W_n) = \{v_i v_{i+1} \mid 1 \leq i \leq n-1\} \cup \{v_n v_1\} \cup \{w v_i \mid 1 \leq i \leq n\}$



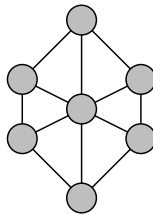
W_3



W_4



W_5



W_6

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

29

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Giới thiệu

Một số đơn đồ thị đặc biệt

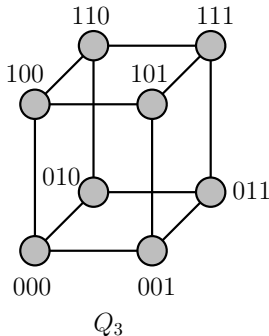
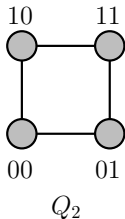
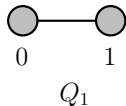


Lý thuyết đồ thị I

Hoàng Anh Đức

Các khối n chiều

Một **khối n chiều** (*n -dimensional cube*), ký hiệu Q_n , là một đồ thị có 2^n đỉnh, mỗi đỉnh được biểu diễn bằng một chuỗi nhị phân độ dài n , và hai đỉnh là liền kề khi và chỉ khi các chuỗi nhị phân biểu diễn chúng khác nhau đúng một bit



Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

30

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Mã trận kề

Mã trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Giới thiệu

Một số đơn đồ thị đặc biệt



Lý thuyết đồ thị I

Hoàng Anh Đức

Bài tập 2

Vẽ các đồ thị sau

(a) K_7

(c) W_7

(b) C_7

(d) Q_4

Bài tập 3

Một đồ thị được gọi là **đồ thị chính quy (regular graph)** nếu các đỉnh của đồ thị có cùng bậc. Ta gọi một đồ thị là d -chính quy nếu nó là đồ thị chính quy trong đó các đỉnh có cùng bậc d . Với các giá trị nào của n thì các đồ thị sau là đồ thị chính quy?

(a) K_n

(c) W_n

(b) C_n

(d) Q_n

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

31

78

Đồ thị hai phần

Giới thiệu



Lý thuyết đồ thị I

Hoàng Anh Đức

Đồ thị hai phần

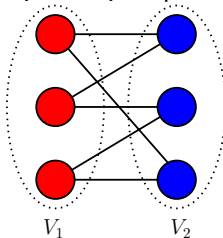
Một đơn đồ thị vô hướng $G = (V, E)$ được gọi là một **đồ thị hai phần (bipartite graph)** nếu tồn tại các tập $V_1 \subseteq V$ và $V_2 \subseteq V$ thỏa mãn $V_1 \neq \emptyset$, $V_2 \neq \emptyset$, $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, và mỗi cạnh của G nối một đỉnh thuộc V_1 và một đỉnh thuộc V_2 . Ta cũng ký hiệu $G = (V_1, V_2, E)$

Định lý 4

Một đơn đồ thị vô hướng $G = (V, E)$ có ít nhất 2 đỉnh là một đồ thị hai phần khi và chỉ khi có một cách tô màu mỗi đỉnh của G bằng hai màu sao cho không có hai đỉnh kề nhau được tô cùng màu

Ví dụ 13

C_6 là một đồ thị hai phần



Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

32

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Đồ thị hai phần

Giới thiệu



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

33

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

- Trong một số tài liệu, các yêu cầu $V_1 \neq \emptyset$ và $V_2 \neq \emptyset$ không được đưa ra trong định nghĩa của đồ thị hai phần (Định nghĩa 1)
- Trong một số tài liệu, Định lý 4 được sử dụng như là định nghĩa của đồ thị hai phần (Định nghĩa 2): Một đồ thị vô hướng G được gọi là **đồ thị hai phần** nếu tồn tại một cách tô màu các đỉnh của G bằng hai màu sao cho không có hai đỉnh kề nhau được tô cùng màu
- Một định nghĩa khác của đồ thị hai phần là như sau (Định nghĩa 3): Một đồ thị vô hướng G được gọi là một **đồ thị hai phần** nếu G không chứa chu trình lẻ nào. Ở đây, một **chu trình lẻ** (**odd cycle**) là một chu trình có số đỉnh là một số lẻ

Đồ thị hai phần

Giới thiệu



Chú ý

Các định nghĩa khác nhau của đồ thị hai phần đôi khi là nguyên nhân của một số mâu thuẫn

Ví dụ, xét đồ thị $G = (V, E)$ chỉ có một đỉnh duy nhất v và không có cạnh nào

- Theo hai định nghĩa đầu tiên, G không phải là một đồ thị hai phần
 - Theo Định nghĩa 1: Giả sử G là đồ thị hai phần. Do $V_1 \neq \emptyset$, $V_2 \neq \emptyset$, $V_1 \cap V_2 = \emptyset$, và $V = V_1 \cup V_2$, tập V phải có ít nhất hai đỉnh phân biệt, mâu thuẫn với giả thiết G chỉ có một đỉnh duy nhất
 - Theo Định nghĩa 2: Giả sử G là đồ thị hai phần. Do G chỉ có một đỉnh duy nhất, không thể tô màu các đỉnh của đồ thị bằng hai màu khác nhau sao cho không có hai đỉnh kề nhau được tô cùng màu, mâu thuẫn với Định nghĩa 2
- Theo Định nghĩa 3: G là một đồ thị hai phần vì G không chứa chu trình nào

Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

34

78

Đồ thị hai phần

Giới thiệu



Lý thuyết đồ thị I

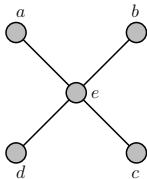
Hoàng Anh Đức

Bài tập 4

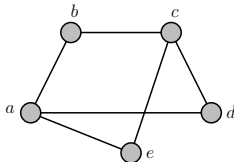
Chứng minh K_n không là đồ thị hai phần với mọi $n \geq 3$. (Gợi ý: Sử dụng phương pháp phản chứng)

Bài tập 5

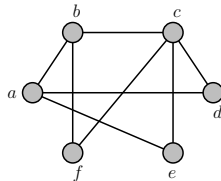
- (a) Chứng minh Định lý 4
- (b) Sử dụng Định lý 4, hãy kiểm tra xem các đồ thị sau có phải đồ thị hai phần hay không



G_1



G_2



G_3

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

35

78

Đồ thị hai phần

Giới thiệu

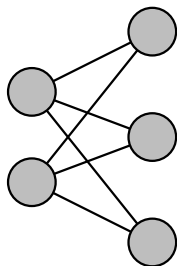


Lý thuyết đồ thị I

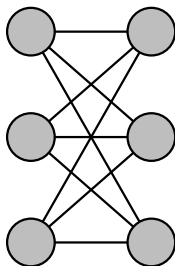
Hoàng Anh Đức

Đồ thị hai phần đầy đủ

Một **đồ thị hai phần đầy đủ** (*complete bipartite graph*) là một đồ thị hai phần $G = (V_1, V_2, E)$ thỏa mãn điều kiện với mọi $v_1 \in V_1$ và $v_2 \in V_2$ ta có $v_1 v_2 \in E$. Nếu $|V_1| = m$ và $|V_2| = n$, ta ký hiệu đồ thị G bằng $K_{m,n}$.



$K_{2,3}$



$K_{3,3}$

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

36

78

Đồ thị hai phần

Giới thiệu



Bài tập 6

Các đồ thị sau có bao nhiêu đỉnh và bao nhiêu cạnh?

- (a) K_n
- (b) C_n
- (c) W_n
- (d) $K_{m,n}$
- (e) Q_n

Bài tập 7

Cho đơn đồ thị vô hướng $G = (V, E)$ có $n \geq 3$ đỉnh. Gọi $H = (W, F)$ là một đồ thị con của G có ít nhất hai đỉnh. Chứng minh rằng nếu G là đồ thị hai phần thì H cũng là đồ thị hai phần. Ngược lại, nếu H là đồ thị hai phần thì G có là đồ thị hai phần hay không? Tại sao?

Bài tập 8

Đồ thị W_n có phải là đồ thị hai phần với mọi $n \geq 3$ hay không? Tại sao? (**Gợi ý:** Sử dụng Bài tập 7 và Bài tập 4)

Bài tập 9

Một đồ thị được gọi là **đồ thị chính quy (regular graph)** nếu các đỉnh của đồ thị có cùng bậc. Chứng minh rằng nếu một đồ thị hai phần $G = (V_1, V_2, E)$ là đồ thị chính quy thì $|V_1| = |V_2|$.

Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

37

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Đồ thị hai phần

Ghép cặp trong đồ thị



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

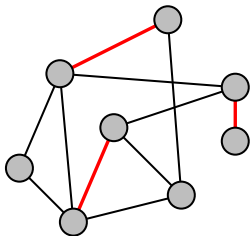
Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

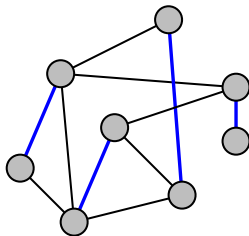
Đếm số đường đi giữa các đỉnh

Cho $G = (V, E)$ là một đơn đồ thị vô hướng

- Một **ghép cặp (matching)** M trong G là một tập con của E thỏa mãn điều kiện không có hai cạnh nào trong M có cùng một đỉnh liên thuộc. Nói cách khác, nếu $uv, st \in M \subseteq E$ thì $\{u, v\} = \{s, t\}$ hoặc $\{u, v\} \cap \{s, t\} = \emptyset$
- Một **ghép cặp cực đại (maximum matching)** trong G là một ghép cặp có số cạnh lớn nhất có thể



M là một ghép cặp



M' là một ghép cặp cực đại

38

78

Đồ thị hai phần

Ghép cặp trong đồ thị



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Mã trận kề

Mã trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

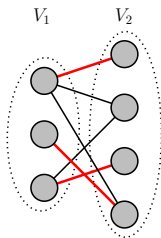
Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

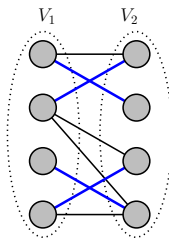
Đếm số đường đi giữa các đỉnh

Cho $G = (V, E)$ là một đơn đồ thị vô hướng

- Ta nói rằng một tập cạnh $W \subseteq E$ **bao phủ (cover)** một tập đỉnh $A \subseteq V$ nếu với mọi đỉnh $u \in A$, tồn tại một cạnh $e \in W$ sao cho e liên thuộc với u , nghĩa là $e = uv$ với đỉnh $v \in V$ nào đó
- Trong một đồ thị hai phần $G = (V_1, V_2, E)$, một **ghép cặp đầy đủ (complete matching)** ứng với V_1 là một ghép cặp $M' \subseteq E$ bao phủ V_1 , và một **ghép cặp hoàn hảo (perfect matching)** là một ghép cặp $M^* \subseteq E$ bao phủ $V = V_1 \cup V_2$



M là một ghép cặp
bao phủ V_1



M' là một ghép cặp
bao phủ V

39

78

Đồ thị hai phần

Ghép cặp trong đồ thị



Lý thuyết đồ thị I

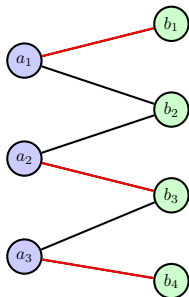
Hoàng Anh Đức

Định lý 5: Định lý Hall (Hall's Marriage Theorem)

Cho $G = (V_1, V_2, E)$ là một đồ thị hai phần. Tồn tại một ghép cặp $M \subseteq E$ bao phủ V_1 khi và chỉ khi với mọi $S \subseteq V_1$, $|S| \leq |N_G(S)|$

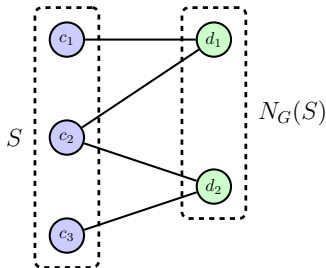
Thỏa mãn điều kiện định lý Hall

$$\forall S \subseteq V_1 : |S| \leq |N_G(S)|$$



Vi phạm điều kiện định lý Hall

$$\exists S \subseteq V_1 : |S| > |N_G(S)|$$



Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

40

78

Đồ thị hai phần

Ứng dụng: Bài toán ghép cặp trong y khoa



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

41

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

- Bài toán phân bổ sinh viên y khoa vào các vị trí thực tập tại các bệnh viện ở Hoa Kỳ
- Trước năm 1950: Quá trình tuyển dụng thực tập hỗn loạn, thiếu hiệu quả
 - Bệnh viện cạnh tranh để có được ứng viên tốt nhất
 - Sinh viên buộc phải quyết định quá sớm trước khi có đầy đủ thông tin
 - Nhiều trường hợp sinh viên và bệnh viện đều không hài lòng với kết quả
- Năm 1952: Thành lập National Resident Matching Program (NRMP), chuyên giải quyết các vấn đề liên quan đến việc phân phối các vị trí thực tập

Đồ thị hai phần

Ứng dụng: Bài toán ghép cặp trong y khoa



Lý thuyết đồ thị I

Hoàng Anh Giang

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

- Mỗi năm, các bác sĩ nộp một danh sách thứ tự các bệnh viện họ muốn nhận lời làm thực tập, và mỗi bệnh viện nộp một danh sách thứ tự các bác sĩ họ muốn nhận làm thực tập
- NRPM tính ra một ghép cặp giữa các bác sĩ và các bệnh viện thỏa mãn yêu cầu về tính **ổn định (stable)**
- Một ghép cặp được gọi là **không ổn định (unstable)** nếu có một cặp bác sĩ α và bệnh viện B đều sẽ thỏa mãn khi được ghép với nhau hơn là với cách ghép cặp hiện tại
 - α được ghép với bệnh viện A nào đó, trong khi cô/anh ấy thích đến bệnh viện B hơn
 - B được ghép với bác sĩ β nào đó, trong khi bệnh viện thích có α đến làm việc hơn

Trong trường hợp này, chúng ta gọi cặp (α, B) là một cặp không ổn định trong ghép cặp.

- Mục tiêu của NRPM là tìm một **ghép cặp ổn định (stable matching)**, nghĩa là một ghép cặp không chứa bất kỳ cặp không ổn định nào

42

78

Đồ thị hai phần

Ví dụ: Ghép cặp ổn định



Lý thuyết đồ thị I

Hoàng Anh Đức

Ví dụ 14 (Ghép cặp ổn định)

Sinh viên và thứ tự ưu tiên:

■ $s_1: h_1 > h_2 > h_3$

■ $s_2: h_2 > h_1 > h_3$

■ $s_3: h_1 > h_3 > h_2$

Bệnh viện và thứ tự ưu tiên:

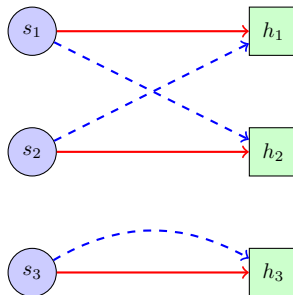
■ $h_1: s_2 > s_1 > s_3$

■ $h_2: s_1 > s_3 > s_2$

■ $h_3: s_3 > s_1 > s_2$

Sinh viên

Bệnh viện



Ghép cặp ổn định: $(s_1, h_1), (s_2, h_2), (s_3, h_3)$
Ghép cặp không ổn định: $(s_1, h_2), (s_2, h_1), (s_3, h_3)$

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

43 Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Đồ thị hai phần

Thuật toán Gale-Shapley



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

- Từ năm 1952, NRMP đã áp dụng thuật toán “Boston Pool” để phân công các thực tập sinh (tên gọi này là do nó đã được sử dụng trước đó bởi một hiệp hội khu vực ở vùng Boston)
- Năm 1962, David Gale và Lloyd Shapley phát triển *thuật toán ghép cặp ổn định (stable matching algorithm)* [Gale and Shapley 1962] bằng cách mô tả và phân tích chính thức một tổng quát hóa của thuật toán Boston Pool và chứng minh rằng nó tính ra một ghép cặp ổn định
- Các thuật toán tương tự từ đó đã được áp dụng cho nhiều trường hợp cần ghép cặp khác, bao gồm tuyển dụng giảng viên ở Pháp, phân công nhiệm vụ cho thủy thủ Hải quân Hoa Kỳ, các chương trình ghép cặp hiến thận, v.v.
- Shapley được trao giải Nobel Kinh tế năm 2012 cho nghiên cứu của ông về ghép cặp ổn định, cùng với Alvin Roth, người đã mở rộng đáng kể công trình của Shapley và phát triển một số ứng dụng trong thực tế. (Gale không nhận được giải thưởng này, vì ông đã qua đời vào năm 2008)

Đồ thị hai phần

Một số thuật toán quan trọng khác liên quan đến ghép cặp



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

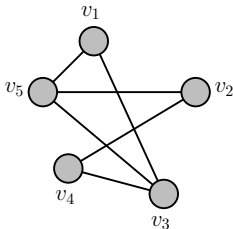
Đếm số đường đi giữa các đỉnh

- **Thuật toán Hungarian (Hungarian algorithm) [Kuhn 1955]:** Tìm ghép cặp có tổng trọng số nhỏ nhất trong đồ thị hai phần có trọng số
- **Thuật toán ghép cặp của Edmonds (Edmonds' blossom algorithm) [Edmonds 1965]:** Tìm ghép cặp cực đại trong đồ thị
- **Thuật toán Hopcroft-Karp [Hopcroft and Karp 1973]:** Tìm ghép cặp cực đại trong đồ thị hai phần
- **Thuật toán Ford-Fulkerson [Ford and Fulkerson 1956]:** Tìm luồng cực đại trong mạng

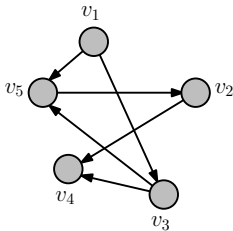
Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Một **danh sách kề (adjacency list)** biểu diễn một đồ thị không có cạnh song song bằng cách liệt kê các đỉnh liền kề với mỗi đỉnh trong đồ thị



Đỉnh	Các đỉnh liền kề
v_1	v_3, v_5
v_2	v_4, v_5
v_3	v_1, v_4, v_5
v_4	v_2, v_3
v_5	v_1, v_2, v_3



Đỉnh bắt đầu	Đỉnh kết thúc
v_1	v_3, v_5
v_2	v_4
v_3	v_4, v_5
v_4	
v_5	v_2



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

46

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Biểu diễn đồ thị và sự đẳng cấu

Ma trận kề



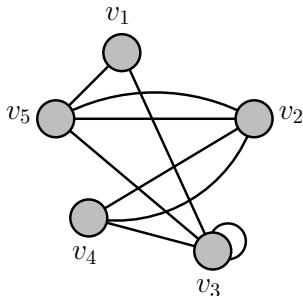
Lý thuyết đồ thị I

Hoàng Anh Đức

Giả sử $G = (V, E)$ là một đồ thị vô hướng có n đỉnh

$v_1, v_2, \dots, v_n$. **Ma trận kề (adjacency matrix)** A của G ứng với thứ tự các đỉnh như trên là một ma trận kích thước $n \times n$ trong đó mỗi phần tử a_{ij} ($1 \leq i, j \leq n$) được định nghĩa như sau

$$a_{ij} = \begin{cases} m_{ij} & \text{nếu có } m_{ij} \text{ cạnh } v_i v_j \\ 0 & \text{nếu } v_i v_j \notin E \end{cases}$$



$$\begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 2 & 2 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

47 Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Biểu diễn đồ thị và sự đẳng cấu

Ma trận kề



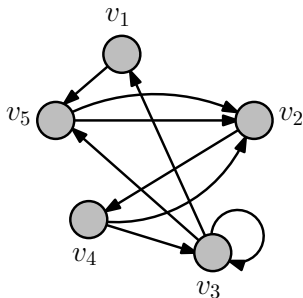
Lý thuyết đồ thị I

Hoàng Anh Đức

Giả sử $G = (V, E)$ là một đồ thị có hướng có n đỉnh

$v_1, v_2, \dots, v_n$. **Ma trận kề (adjacency matrix)** A của G ứng với thứ tự các đỉnh như trên là một ma trận kích thước $n \times n$ trong đó mỗi phần tử a_{ij} ($1 \leq i, j \leq n$) được định nghĩa như sau

$$a_{ij} = \begin{cases} m_{ij} & \text{nếu có } m_{ij} \text{ cạnh } (v_i, v_j) \\ 0 & \text{nếu } (v_i, v_j) \notin E \end{cases}$$



$$\begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

48

78

Biểu diễn đồ thị và sự đẳng cấu

Ma trận liên thuộc



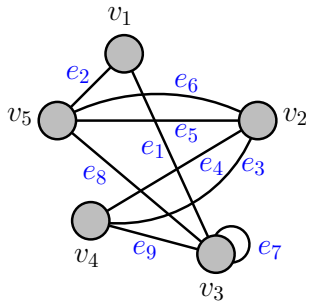
Lý thuyết đồ thị I

Hoàng Anh Đức

Giả sử $G = (V, E)$ là một đồ thị vô hướng có n đỉnh

$v_1, v_2, \dots, v_n$ và m cạnh $e_1, e_2, \dots, e_m$. **Ma trận liên thuộc (incidence matrix) A** của G tương ứng với thứ tự các đỉnh và cạnh như trên là một ma trận kích thước $n \times m$ trong đó các phần tử a_{ij} ($1 \leq i \leq n$ và $1 \leq j \leq m$) được định nghĩa như sau

$$a_{ij} = \begin{cases} 1 & \text{nếu cạnh } e_j \text{ liên thuộc với đỉnh } v_i \\ 0 & \text{ngược lại} \end{cases}$$



$$\begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

49

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Biểu diễn đồ thị và sự đẳng cấu

Ma trận liên thuộc

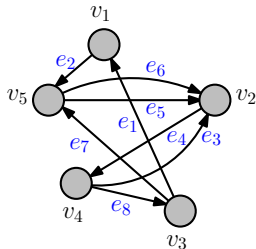


Lý thuyết đồ thị I

Hoàng Anh Đức

Giả sử $G = (V, E)$ là một đồ thị có hướng có n đỉnh $v_1, v_2, \dots, v_n$ và m cạnh $e_1, e_2, \dots, e_m$. Giả sử G không chứa khuyên. **Ma trận liên thuộc (incidence matrix)** A của G tương ứng với thứ tự các đỉnh và cạnh như trên là một ma trận kích thước $n \times m$ trong đó các phần tử a_{ij} ($1 \leq i \leq n$ và $1 \leq j \leq m$) được định nghĩa như sau

$$a_{ij} = \begin{cases} 1 & \text{nếu cạnh } e_j \text{ có đỉnh đầu là } v_i \\ -1 & \text{nếu cạnh } e_j \text{ có đỉnh cuối là } v_i \\ 0 & \text{ngược lại} \end{cases}$$



	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_1	-1	1	0	0	0	0	0	0
v_2	0	0	-1	1	-1	-1	0	0
v_3	1	0	0	0	0	0	1	-1
v_4	0	0	1	-1	0	0	0	1
v_5	0	-1	0	0	1	1	-1	0

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

50

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Biểu diễn đồ thị và sự đẳng cấu

Sự đẳng cấu giữa các đồ thị

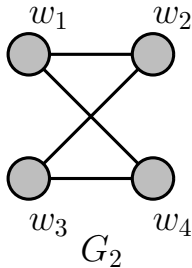
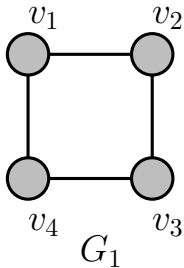


Lý thuyết đồ thị I

Hoàng Anh Đức

Sự đẳng cấu

Hai đồ thị vô hướng $G_1 = (V_1, E_1)$ và $G_2 = (V_2, E_2)$ là **đẳng cấu** (*isomorphic*), ký hiệu $G_1 \simeq G_2$ hoặc $G_1 \cong G_2$, nếu tồn tại một song ánh $f : V_1 \rightarrow V_2$ thỏa mãn điều kiện: với mọi đỉnh $u, v \in V_1$, $uv \in E_1$ khi và chỉ khi $f(u)f(v) \in E_2$



Hình: $G_1 \simeq G_2$ do tồn tại song ánh $f : V_1 \rightarrow V_2$ định nghĩa bởi $f(v_i) = w_i$ ($1 \leq i \leq 4$) thỏa mãn điều kiện đề ra

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

51

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Biểu diễn đồ thị và sự đẳng cấu

Sự đẳng cấu giữa các đồ thị



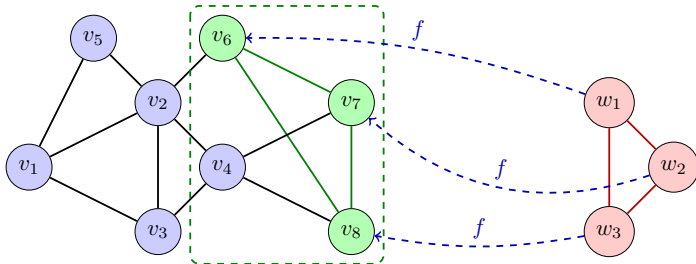
Lý thuyết đồ thị I

Hoàng Anh Đức

Chú ý

Với hai đồ thị phân biệt $G = (V, E)$ và $H = (W, F)$, khi nói **đồ thị H là đồ thị con của đồ thị G** hoặc **G chứa H** , có thể hiểu rằng tồn tại một đồ thị $H' = (W', F')$ thỏa mãn $W' \subseteq V$, $F' \subseteq E$ và $H' \simeq H$. Tương tự với các khái niệm đồ thị con bao trùm, đồ thị con thực sự, đồ thị con cảm sinh, v.v.

$H' = (W', F')$ với $W' \subseteq V$, $F' \subseteq E$



$G = (V, E)$

$H' \simeq H$ (Đẳng cấu)

$H = (W, F)$

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Mã trận kề

Mã trận liên thuộc

52

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Biểu diễn đồ thị và sự đẳng cấu

Sự đẳng cấu giữa các đồ thị



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

- Một số tính chất hiển nhiên mà các đồ thị đẳng cấu

$G_1 = (V_1, E_1)$ và $G_2 = (V_2, E_2)$ cần có

- $|V_1| = |V_2|$

- $|E_1| = |E_2|$

- Với mỗi d , số đỉnh bậc d trong G_1 bằng số đỉnh bậc d trong G_2

- v.v...

- Thông thường, *việc kiểm tra tất cả các song ánh* có thể giữa hai tập đỉnh của hai đồ thị G_1, G_2 để xác định xem chúng có đẳng cấu hay không là *rất khó*: có $n!$ song ánh giữa hai đồ thị n đỉnh

- Đến hiện tại, *chưa biết* có hay không một *thuật toán trong thời gian đa thức* để kiểm tra xem hai đồ thị là đẳng cấu hay không

- Thêm nữa, một *bài toán mở trong hàng thập kỷ* là *thiết kế một thuật toán với độ phức tạp trong thời gian xấu nhất tốt hơn thời gian $O(2^{\sqrt{n}})$* . Một bước *đột phá* là kết quả của László

Babai [Babai 2016]: thuật toán trong thời gian $2^{O((\log n)^k)}$ với hằng số $k > 1$ cố định nào đó. Thời gian chạy của thuật toán này "tốt hơn" $O(2^{\sqrt{n}})$ nhưng "tệ hơn" thời gian đa thức. Harald Andrés Helfgott chứng minh rằng có thể lấy hằng số $k = 3$

53

78

Biểu diễn đồ thị và sự đẳng cấu

Sự đẳng cấu giữa các đồ thị



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

- Để chứng minh hai đồ thị là *không đẳng cấu*, chúng ta thường tìm một tính chất mà chỉ một trong hai đồ thị có. Một tính chất như thế được gọi là một *bất biến đồ thị (graph invariant)* (ví dụ như số các đỉnh có bậc cho trước nào đó, danh sách bậc các đỉnh của đồ thị, v.v...)
- Trên thực tế, có những thuật toán *rất hiệu quả cho phần lớn các đồ thị xuất hiện trong thực hành*
 - Ví dụ, phần mềm “nauty” của Brendan McKay (cho đồ thị vô hướng) và “Traces” của Adolfo Piperno (xem <https://pallini.di.uniroma1.it/>)

54

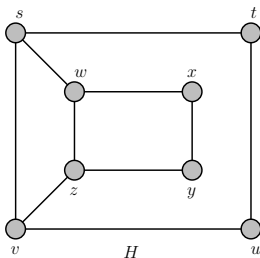
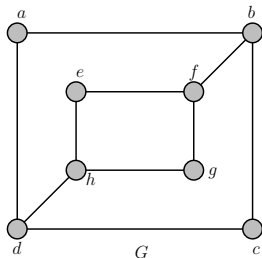
78

Biểu diễn đồ thị và sự đẳng cấu

Sự đẳng cấu giữa các đồ thị



Ví dụ 15



G và H không đẳng cấu

- Do $\deg(a) = 2$, nếu tồn tại một đẳng cấu giữa G và H , a phải tương ứng với một trong bốn đỉnh bậc 2 của H : t, u, x , hoặc y
- Tuy nhiên, mỗi đỉnh trong bốn đỉnh t, u, x, y đều liên kết với một đỉnh bậc hai, trong khi a không thỏa mãn tính chất này trong G

Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

55 Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Biểu diễn đồ thị và sự đẳng cấu

Sự đẳng cấu giữa các đồ thị

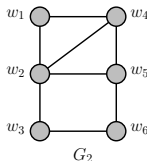
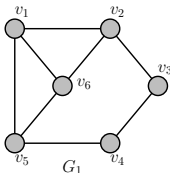
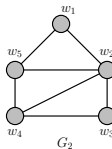
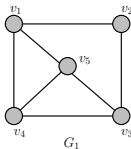
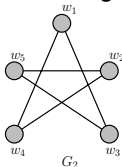
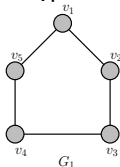


Lý thuyết đồ thị I

Hoàng Anh Đức

Bài tập 10

Các cặp đồ thị sau có đẳng cấu hay không? Vì sao?



Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Mã trận kề

Mã trận liên thuộc

56

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Bài tập 11

Giả sử G và H là các đơn đồ thị thỏa mãn $G \simeq H$. Chứng minh rằng $\overline{G} \simeq \overline{H}$

Biểu diễn đồ thị và sự đẳng cấu

Sự đẳng cấu giữa các đồ thị



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

57 Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

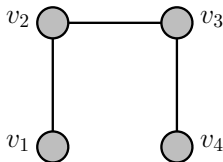
Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Bài tập 12

Một đơn đồ thị G được gọi là *tự bù* (self-complementary) nếu $G \simeq \overline{G}$.

(a) Chứng minh rằng đồ thị sau là một đồ thị tự bù



(b) Tìm một đồ thị tự bù có 5 đỉnh.

Tính liên thông trong đồ thị

Đường đi



Lý thuyết đồ thị I

Hoàng Anh Đức

Đường đi (vô hướng)

Cho $G = (V, E)$ là một đồ thị vô hướng và n là một số nguyên dương. **Đường đi (path)** độ dài n từ đỉnh u đến đỉnh v trong G là một dãy các cạnh $e_1, e_2, \dots, e_n$ của đồ thị thỏa mãn điều kiện tồn tại một dãy các đỉnh $v_0, v_1, v_2, \dots, v_{n-1}, v_n$ sao cho $v_0 = u$, $v_n = v$, và e_i có các đầu mút v_{i-1} và v_i , với mọi $i \in \{1, 2, \dots, n\}$

- Ta nói rằng đường đi bắt đầu với u và kết thúc với v
- **Độ dài (length)** của một đường đi vô hướng là số cạnh của đường đi đó
- Một đường đi độ dài $n \geq 1$ được gọi là một **chu trình (circuit hoặc cycle)** nếu nó bắt đầu và kết thúc ở cùng một đỉnh
- Khi G **không có các cạnh song song và khuyên**, mỗi đường đi có thể được xác định một cách duy nhất thông qua các đỉnh của nó, và do đó ta có thể ký hiệu một đường đi bằng dãy các đỉnh của nó $v_0, v_1, \dots, v_n$

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

58

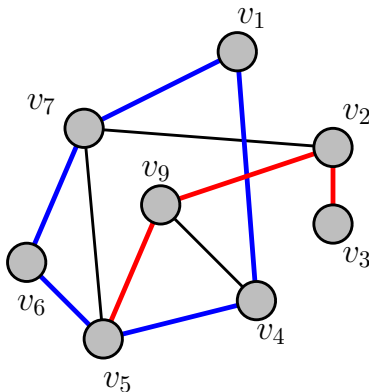
78

Tính liên thông trong đồ thị

Đường đi



Ví dụ 16



Hình: v_5v_9, v_9v_2, v_2v_3 (hoặc v_5, v_9, v_2, v_3) là một đường đi độ dài 3 và $v_1v_4, v_4v_5, v_5v_6, v_6v_7, v_7v_1$ (hoặc $v_1, v_4, v_5, v_6, v_7, v_1$) là một chu trình độ dài 5

Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

59

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Tính liên thông trong đồ thị

Đường đi



Lý thuyết đồ thị I

Hoàng Anh Đức

Đường đi (có hướng)

Cho $G = (V, E)$ là một đồ thị có hướng và n là một số nguyên dương. **Đường đi (path)** độ dài n từ đỉnh u đến đỉnh v trong G là một dãy các cung $e_1, e_2, \dots, e_n$ của đồ thị thỏa mãn điều kiện tồn tại một dãy các đỉnh $v_0, v_1, v_2, \dots, v_{n-1}, v_n$ sao cho $v_0 = u, v_n = v$, và e_i có đỉnh đầu v_{i-1} và đỉnh cuối v_i , với mọi $i \in \{1, 2, \dots, n\}$

- Ta nói rằng đường đi bắt đầu với u và kết thúc với v
- **Độ dài (length)** của một đường đi có hướng là số cung của đường đi đó
- Một đường đi độ dài $n \geq 1$ được gọi là một **chu trình (circuit hoặc cycle)** nếu nó bắt đầu và kết thúc ở cùng một đỉnh
- Khi G **không có các cạnh song song và khuyên**, mỗi đường đi có thể được xác định một cách duy nhất thông qua các đỉnh của nó, và do đó ta có thể ký hiệu một đường đi bằng dãy các đỉnh của nó $v_0, v_1, \dots, v_n$

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

60

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

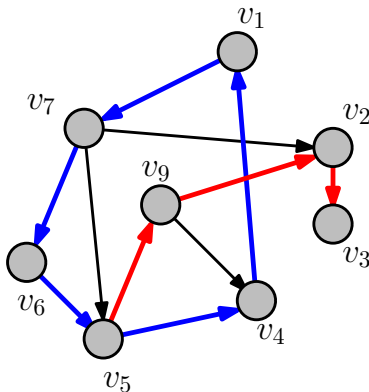
78

Tính liên thông trong đồ thị

Đường đi



Ví dụ 17



Hình: $(v_5, v_9), (v_9, v_2), (v_2, v_3)$ (hoặc v_5, v_9, v_2, v_3) là một đường đi độ dài 3 và $(v_1, v_7), (v_7, v_6), (v_6, v_5), (v_5, v_4), (v_4, v_1)$ (hoặc $v_1, v_7, v_6, v_5, v_4, v_1$) là một chu trình độ dài 5

Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

61

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Tính liên thông trong đồ thị

Đường đi



Lý thuyết đồ thị I

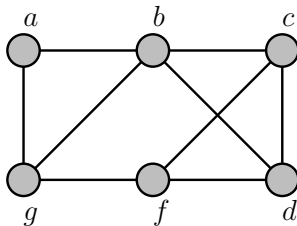
Hoàng Anh Đức

Một **đường đi đơn (simple path)** là một đường đi không chứa cùng một cạnh (cung) nhiều hơn một lần

Bài tập 13

Hãy tìm trong đồ thị ở hình bên

- (a) Một đường đi có độ dài n với $n \in \{1, 2, \dots, 7\}$
- (b) Một đường đi đơn có độ dài n với $n \in \{1, 2, \dots, 7\}$
- (c) Một chu trình có độ dài n với $n \in \{3, \dots, 7\}$



Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Mã trận kề

Mã trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

62 Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Tính liên thông trong đồ thị

Liên thông trong đồ thị vô hướng



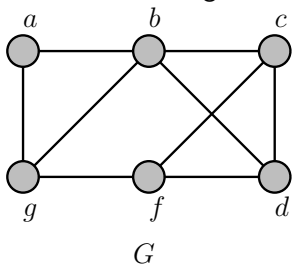
Lý thuyết đồ thị I

Hoàng Anh Đức

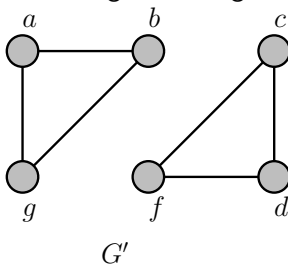
Một đồ thị vô hướng $G = (V, E)$ được gọi là **liên thông** (*connected*) nếu có đường đi giữa mọi cặp đỉnh phân biệt của G . Ngược lại, nếu không tồn tại đường đi giữa một cặp đỉnh phân biệt nào đó trong G , ta gọi G là đồ thị **không liên thông** (*disconnected*).

- Đồ thị có chính xác một đỉnh luôn được coi là liên thông
- Đồ thị không có đỉnh nào (**đồ thị rỗng** (*null graph*)) không được coi là liên thông

Liên thông



Không liên thông



Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

63

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Tính liên thông trong đồ thị

Liên thông trong đồ thị vô hướng



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

64

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

- Một **thành phần liên thông (connected component)** $H = (V', E')$ của G là một **đồ thị con liên thông cực đại** của G , nghĩa là, H là một đồ thị con liên thông của G và với mọi đồ thị con liên thông K khác của G , H không là đồ thị con thực sự của K
- **Hợp (union)** của hai đồ thị $G_1 = (V_1, E_1)$ và $G_2 = (V_2, E_2)$ là một đồ thị $G = (V, E)$ có tập đỉnh $V = V_1 \cup V_2$ và tập cạnh $E = E_1 \cup E_2$. Ta cũng viết $G = G_1 \cup G_2$
- Một **đồ thị không liên thông** G có thể được xem như là **hợp của hai hay nhiều đồ thị con liên thông trong đó không có đỉnh chung nào giữa mỗi cặp đồ thị con này**. Các đồ thị con này chính là các thành phần liên thông của G
- G là đồ thị **liên thông** khi và chỉ khi G **có chính xác một thành phần liên thông**

Tính liên thông trong đồ thị

Liên thông trong đồ thị vô hướng



Mệnh đề 6

Cho $G = (V, E)$ là một đồ thị vô hướng liên thông có ít nhất hai đỉnh. Với hai đỉnh bất kỳ $u, v \in V$ của G , tồn tại một đường đi đơn giữa u và v

Chứng minh.

- Do G liên thông, luôn tồn tại một đường đi giữa hai đỉnh u, v . Gọi $P = e_1, e_2, \dots, e_k$ là một đường đi có độ dài nhỏ nhất trong số tất cả các đường đi giữa u và v . Ta chứng minh P là một đường đi đơn
- Giả sử P không phải đường đi đơn. Suy ra, tồn tại i, j thỏa mãn $0 \leq i < j \leq k$ và $e_i = e_j$. Do đó, $P' = e_1, e_2, \dots, e_i, e_{j+1}, \dots, e_k$ là một đường đi giữa u và v và P' có độ dài nhỏ hơn độ dài k của P . Điều này mâu thuẫn với định nghĩa của P

Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ
Định nghĩa và khái niệm
Đồ thị mới từ đồ thị cũ
Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu
Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề
Ma trận kề
Ma trận liên thuộc
Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi
Liên thông trong đồ thị vô hướng
Liên thông trong đồ thị có hướng
Đường đi và sự đẳng cấu
Đếm số đường đi giữa các đỉnh

65



78

Tính liên thông trong đồ thị

Liên thông trong đồ thị vô hướng

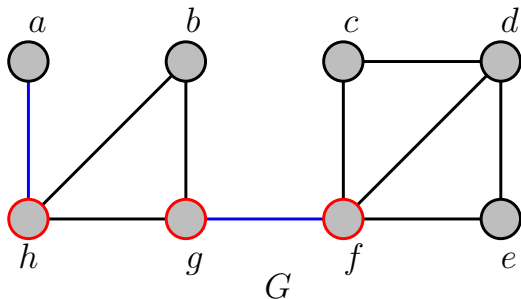


Lý thuyết đồ thị I

Hoàng Anh Đức

Cho đồ thị vô hướng $G = (V, E)$

- Một đỉnh $v \in V$ được gọi là **đỉnh cắt (cut vertex)** hoặc **điểm khớp (articulation point)** nếu $G - v$ có nhiều thành phần liên thông hơn G
- Một cạnh $e \in E$ được gọi là **cạnh cắt (cut edge)** hoặc **cầu (bridge)** nếu $G - e$ có nhiều thành phần liên thông hơn G



Hình: Các đỉnh cắt của G là f, g, h . Các cạnh cắt của G là ah, gf

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

66

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Tính liên thông trong đồ thị

Liên thông trong đồ thị vô hướng



Lý thuyết đồ thị I

Hoàng Anh Đức

Bài tập 14

Một đồ thị không có đỉnh cắt nào được gọi là **đồ thị không thể tách rời (nonseparable graph)**. Chứng minh rằng mỗi đồ thị sau là đồ thị không thể tách rời

(a) C_n với $n \geq 3$

(c) $K_{m,n}$ với $m \geq 2$ và $n \geq 2$

(b) W_n với $n \geq 3$

(d) Q_n với $n \geq 2$

Bài tập 15

Chứng minh rằng nếu G là đơn đồ thị vô hướng có chính xác hai đỉnh bậc lẻ u, v thì các đỉnh này phải thuộc cùng một thành phần liên thông của G

Bài tập 16

Chứng minh rằng một đồ thị vô hướng liên thông bất kỳ gồm $n \geq 1$ đỉnh có ít nhất $n - 1$ cạnh. (**Gợi ý:** Quy nạp mạnh theo số đỉnh n của đồ thị.)

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

67

78

Tính liên thông trong đồ thị

Liên thông trong đồ thị vô hướng



Lý thuyết đồ thị I

Hoàng Anh Đức

Bài tập 17

Cho $G = (V, E)$ là một đơn đồ thị vô hướng liên thông gồm $n \geq 1$ đỉnh và $G \neq K_n$. Chứng minh rằng luôn tồn tại một tập các đỉnh V' sao cho $G - V'$ là đồ thị không liên thông

- Tập đỉnh V' của một đơn đồ thị vô hướng liên thông G thỏa mãn điều kiện ở Bài tập 17 được gọi là một **tập phân tách (separating set (of vertices))** của G
- **Số liên thông đỉnh (vertex connectivity)** của G , ký hiệu $\kappa(G)$, là **số đỉnh nhỏ nhất cần bỏ đi từ G** để thu được một đồ thị con G' **không liên thông** hoặc **chỉ có một đỉnh**
 - $\kappa(G) = 0$ nếu G không liên thông hoặc chỉ có một đỉnh
 - $\kappa(G)$ là số phần tử nhỏ nhất trong một tập phân tách (nếu có) của G
- G là **k -liên thông (k -connected)** nếu $\kappa(G) \geq k$
 - Nếu G là k -liên thông thì cũng là j -liên thông với mọi $0 \leq j \leq k$
 - Nếu **xóa đi tối đa $k - 1$ đỉnh bất kỳ** từ G thì đồ thị thu được luôn là đồ thị liên thông

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

68

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Tính liên thông trong đồ thị

Liên thông trong đồ thị vô hướng



Lý thuyết đồ thị I

Hoàng Anh Đức

Bài tập 18

Cho $G = (V, E)$ là một đơn đồ thị vô hướng liên thông gồm $n \geq 2$ đỉnh. Chứng minh rằng luôn tồn tại một tập cạnh E' sao cho $G - E'$ là một đồ thị không liên thông

- Tập cạnh E' của một đơn đồ thị vô hướng liên thông G thỏa mãn điều kiện ở Bài tập 18 được gọi là một **tập cạnh phân tách (separating set of edges)** của G
- **Số liên thông cạnh (edge connectivity)** của G , ký hiệu $\lambda(G)$, là **số cạnh nhỏ nhất cần bỏ đi từ G để thu được một đồ thị con G' không liên thông hoặc chỉ có một đỉnh**
 - $\lambda(G) = 0$ nếu G không liên thông hoặc chỉ có một đỉnh
 - $\lambda(G)$ là số phần tử nhỏ nhất trong một tập cạnh phân tách (nếu có) của G
- G là **k -liên thông cạnh (k -edge connected)** nếu $\lambda(G) \geq k$
 - Nếu G là k -liên thông cạnh thì cũng là j -liên thông cạnh với mọi $0 \leq j \leq k$
 - Nếu **xóa đi tối đa $k - 1$ cạnh bất kỳ** từ G , đồ thị thu được luôn là đồ thị liên thông

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

69

78

Tính liên thông trong đồ thị

Liên thông trong đồ thị vô hướng



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

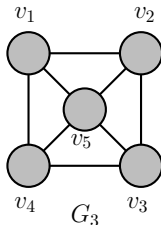
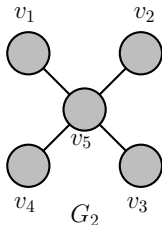
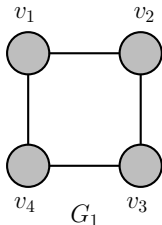
Để chứng minh $\kappa(G) = k$ với đơn đồ thị vô hướng liên thông $G = (V, E)$, cần chứng minh:

- (1) Tồn tại một tập $W \subseteq V$ gồm k đỉnh thỏa mãn $G - W$ không liên thông hoặc chỉ có 1 đỉnh
- (2) Với mọi tập $X \subseteq V$ gồm ℓ đỉnh ($\ell < k$), $G - X$ là đồ thị liên thông

Tương tự với $\lambda(G)$.

Bài tập 19

Xác định $\kappa(G_i)$ và $\lambda(G_i)$ trong các đồ thị G_i với $i \in \{1, 2, 3\}$ sau



70

78

Tính liên thông trong đồ thị

Liên thông trong đồ thị vô hướng



Lý thuyết đồ thị I

Hoàng Anh Đức

Bài tập 20

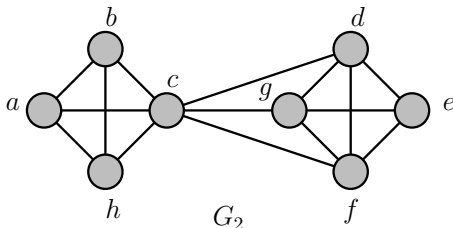
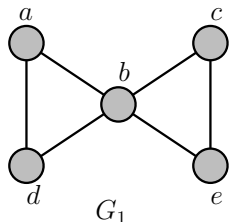
Chứng minh rằng với mọi đồ thị vô hướng liên thông $G = (V, E)$

$$\kappa(G) \leq \min_{v \in V} \deg_G(v) \quad (1)$$

$$\lambda(G) \leq \min_{v \in V} \deg_G(v) \quad (2)$$

Bài tập 21

Với mỗi đồ thị G_i ($i \in \{1, 2\}$) sau, tìm $\kappa(G)$, $\lambda(G)$, và $\min_{v \in V} \deg(v)$



Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

71

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

78

Tính liên thông trong đồ thị

Liên thông trong đồ thị có hướng



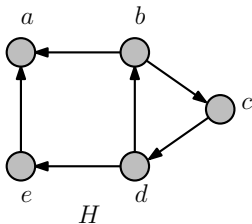
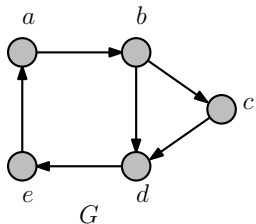
Lý thuyết đồ thị I

Hoàng Anh Đức

Cho $G = (V, E)$ là một đồ thị có hướng

- G được gọi là **liên thông mạnh (strongly connected)** nếu với mỗi cặp đỉnh $u, v \in V$, tồn tại một đường đi có hướng từ u đến v và một đường đi có hướng từ v đến u
- G được gọi là **liên thông yếu (weakly connected)** nếu đồ thị vô hướng thu được bằng cách bỏ qua hướng của các cung của G là một đồ thị liên thông

Ví dụ 18



Hình: G là đồ thị liên thông mạnh. H không là đồ thị liên thông mạnh nhưng là đồ thị liên thông yếu

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu
Đếm số đường đi giữa các đỉnh

72

78

Tính liên thông trong đồ thị

Liên thông trong đồ thị có hướng



Lý thuyết đồ thị I

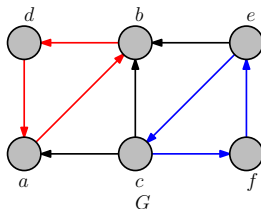
Hoàng Anh Đức

Cho $G = (V, E)$ là một đồ thị có hướng

- Một **thành phần liên thông mạnh (strongly connected component)** của G là một đồ thị con liên thông mạnh cực đại H của G , nghĩa là, H là một đồ thị con liên thông mạnh của G và không là đồ thị con thực sự của bất kỳ đồ thị con liên thông mạnh nào khác

Ví dụ 19

- G không là đồ thị liên thông mạnh
- Đồ thị $G_1 = (V_1, E_1)$ với $V_1 = \{a, b, d\}$ và $E_1 = \{(a, b), (b, d), (d, a)\}$ là một thành phần liên thông mạnh của G
- Đồ thị $G_2 = (V_2, E_2)$ với $V_2 = \{c, e, f\}$ và $E_2 = \{(c, f), (f, e), (e, c)\}$ là một thành phần liên thông mạnh của G



Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

73

78

Tính liên thông trong đồ thị

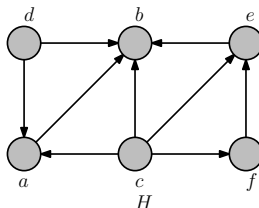
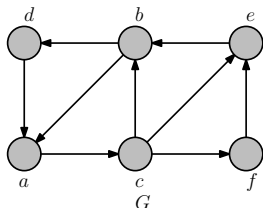
Liên thông trong đồ thị có hướng



Lý thuyết đồ thị I

Hoàng Anh Đức

- Một **đồ thị có hướng không có chu trình (directed acyclic graph – DAG)** là một đồ thị có hướng không chứa khuyên hoặc chu trình có hướng.



Hình: G là một đồ thị có hướng và có chu trình. H là một đồ thị có hướng và không có chu trình

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

74

78

Tính liên thông trong đồ thị

Đường đi và sự đẳng cấu



Lý thuyết đồ thị I

Hoàng Anh Đức

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu
Đếm số đường đi giữa các đỉnh

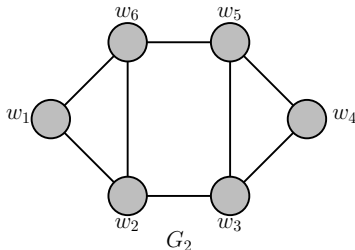
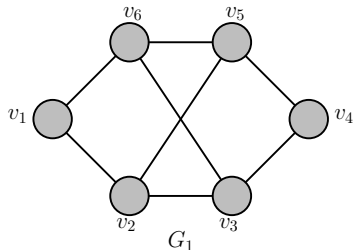
- **Nhắc lại:** Để chứng minh hai đồ thị là *không đẳng cấu*, chúng ta thường tìm một tính chất mà chỉ một trong hai đồ thị có. Một tính chất như thế được gọi là một *bất biến đồ thị (graph invariant)*

- số các đỉnh có bậc cho trước nào đó
- danh sách bậc các đỉnh của đồ thị

- Một bất biến đồ thị hữu ích là *sự tồn tại của các chu trình đơn với độ dài $k \geq 3$*

Bài tập 22

Các đồ thị sau có đẳng cấu không? Vì sao?



75

78

Tính liên thông trong đồ thị

Đường đi và sự đẳng cấu



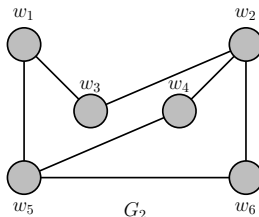
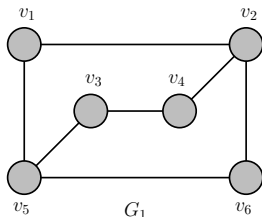
Lý thuyết đồ thị I

Hoàng Anh Đức

- Chúng ta cũng có thể sử dụng đường đi để tìm các ánh xạ giữa hai đồ thị đẳng cấu

Bài tập 23

Các đồ thị sau có đẳng cấu không? Vì sao?



Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Tính liên thông trong đồ thị

Đếm số đường đi giữa các đỉnh



Lý thuyết đồ thị I

Hoàng Anh Đức

Định lý 7

Cho G là một đồ thị với ma trận kề A tương ứng với thứ tự các đỉnh $v_1, v_2, \dots, v_n$. Số các đường đi khác nhau độ dài r từ v_i tới v_j , trong đó r là một số nguyên dương, bằng giá trị của phần tử (i, j) của ma trận A^r .

Chứng minh.

Ta chứng minh Định lý bằng quy nạp theo r

- **Bước cơ sở:** Theo định nghĩa ma trận kề, Định lý 7 đúng với $r = 1$
- **Bước quy nạp:** Giả sử Định lý 7 đúng với mọi $1 \leq r \leq k$. Ta chứng minh Định lý 7 đúng với $r = k + 1$, tức là, số các đường đi khác nhau độ dài $k + 1$ từ v_i tới v_j bằng giá trị của phần tử (i, j) của A^{k+1} .
 - Một đường đi độ dài $k + 1$ từ v_i đến v_j được tạo thành bởi một đường đi độ dài k từ v_i đến v_ℓ nào đó, và cạnh $\{v_\ell, v_j\}$.

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

77

78

Tính liên thông trong đồ thị

Đếm số đường đi giữa các đỉnh



Lý thuyết đồ thị I

Hoàng Anh Đức

Bài tập 24

Tìm số đường đi độ dài n giữa hai đỉnh phân biệt của K_4 với n bằng

(a) 2

(b) 3

(c) 4

(d) 5

Bài tập 25

Tìm số đường đi đơn độ dài n giữa hai đỉnh phân biệt của K_4 với n bằng

(a) 2

(b) 3

(c) 4

(d) 5

Giới thiệu

Định nghĩa đồ thị và một số ví dụ

Định nghĩa và khái niệm

Đồ thị mới từ đồ thị cũ

Một số đơn đồ thị đặc biệt

Đồ thị hai phần

Giới thiệu

Ghép cặp trong đồ thị

Biểu diễn đồ thị và sự đẳng cấu

Danh sách kề

Ma trận kề

Ma trận liên thuộc

Sự đẳng cấu giữa các đồ thị

Tính liên thông trong đồ thị

Đường đi

Liên thông trong đồ thị vô hướng

Liên thông trong đồ thị có hướng

Đường đi và sự đẳng cấu

Đếm số đường đi giữa các đỉnh

Part I

Phụ lục



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Một số lỗi thường gặp



Lý thuyết đồ thị I

Hoàng Anh Đức

Chú ý

Tham khảo từ tài liệu “Common Mistakes in Discrete Mathematics” (https://highered.mheducation.com/sites/dl/free/125967651x/1106131/Common_Mistakes_in_Discrete_Math.pdf)

2 Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

(a) Nhầm lẫn giữa một số thuật ngữ trong lý thuyết đồ thị

- Chẳng hạn như sự khác biệt giữa đường đi (path) và đường đi đơn (simple path). Đây là một trong những trường hợp cần phải ghi nhớ
- Việc tạo ra bảng thuật ngữ riêng trên các thẻ ghi chú hoặc trong một tệp máy tính có thể hữu ích

(b) Đếm thừa cạnh trong đồ thị do quên chia 2 khi cộng bậc của các đỉnh

- Mỗi cạnh được đếm hai lần, một lần cho mỗi đầu.

Một số lỗi thường gặp (tiếp)



Lý thuyết đồ thị I

Hoàng Anh Đức

(c) Không chắc chắn về việc nên sử dụng mô hình đồ thị nào

- Nếu mỗi quan hệ giữa các đối tượng trong tình huống bạn đang mô hình hóa có tính đối xứng, thì đồ thị vô hướng có thể phù hợp; ngược lại, đồ thị có hướng thường phù hợp hơn
- Ví dụ: đường cao tốc nối các thành phố lớn có thể đi theo cả hai hướng, nên đồ thị vô hướng là phù hợp cho mô hình này
- Mỗi quan hệ săn mồi-con mồi giữa các loài động vật rõ ràng không có tính đối xứng, nên đồ thị có hướng dường như đúng hơn trong trường hợp này

(d) Sai lầm khi cho rằng nếu hai đồ thị có nhiều thuộc tính giống nhau (bất biến như số đỉnh, số cạnh, v.v.) thì chúng phải đẳng cấu

- Ví dụ: đồ thị $K_{3,3}$ và đồ thị K_6 đều có 6 đỉnh và 9 cạnh, nhưng chúng không đẳng cấu với nhau

(e) Nhầm lẫn về các tính chất của đồ thị hai phần

3 Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

Một số lỗi thường gặp (tiếp)



Lý thuyết đồ thị I

Hoàng Anh Đức

4 Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

- Cho rằng nếu G là một đồ thị hai phần thì nó luôn là đồ thị liên thông
- Cho rằng nếu G có chu trình thì nó không thể là đồ thị hai phần

Đồ thị web và một số mạng lưới tương tự

Đồ thị web $\mathcal{W}$



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

5

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

- Xuất hiện sau sự ra đời của Internet vào đầu những năm 1980
- Mỗi trang web được coi là một đỉnh trong đồ thị $\mathcal{W}$, mỗi liên kết giữa các trang web được coi là một cạnh
- Mặc dù các siêu liên kết (hyperlinks) là các cạnh có hướng, trong một số trường hợp $\mathcal{W}$ có thể được xem như một đồ thị vô hướng
- Đồ thị web $\mathcal{W}$ là một đồ thị thực tế, phát triển liên tục theo thời gian
- Ngoài ra, có những trang web động (ví dụ như đồng hồ trực tuyến) và số lượng các trang như vậy có thể được xem là vô hạn
- Do đó, đồ thị web $\mathcal{W}$ có thể được xem là hữu hạn hoặc vô hạn

Đồ thị web và một số mạng lưới tương tự

Đồ thị web $\mathcal{W}$



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

6 Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

Một số tính chất của đồ thị web $\mathcal{W}$ [Zverovich 2021]:

- $\mathcal{W}$ là một đồ thị *khổng lồ* với hàng tỷ đỉnh (ví dụ, xem các cơ sở dữ liệu về đồ thị web ở <https://webgraph.di.unimi.it/>)
- $\mathcal{W}$ là một đồ thị *tương đối thưa thớt (sparse)* vì số cạnh trong $\mathcal{W}$ ít hơn nhiều so với số cạnh có thể có. Cụ thể hơn, bậc trung bình của một đỉnh cố định trong $\mathcal{W}$ không vượt quá một hằng số [Baldi, Frasconi, and Smyth 2003]

$$\text{Bậc trung bình của } \mathcal{W} = \frac{\text{tổng bậc các đỉnh trong } \mathcal{W}}{\text{số đỉnh của } \mathcal{W}}$$

Đồ thị web và một số mạng lưới tương tự

Đồ thị web $\mathcal{W}$



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

7 Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

Một số tính chất của đồ thị web $\mathcal{W}$ [Zverovich 2021]:

- Đồ thị web có nhiều “cộng đồng”. Một **cộng đồng** (*community*) là một tập hợp các trang web có cùng mối liên hệ nào đó. Ví dụ, một cộng đồng có thể là một nhóm các trang web liên quan đến một chủ đề cụ thể như thể thao, du lịch, v.v.
 - Có nhiều cách tiếp cận để định nghĩa “cộng đồng” theo ngôn ngữ toán học
 - Một cách là định nghĩa cộng đồng như một đồ thị con có số cạnh nội bộ (nối giữa các đỉnh trong cộng đồng) nhiều hơn số cạnh bên ngoài (nối giữa các đỉnh trong cộng đồng và các đỉnh bên ngoài cộng đồng)
 - Một định nghĩa khác dựa trên **các đồ thị con hai phần có hướng dày đặc** (*dense directed bipartite subgraphs*) được gọi là **lõi hai phần** (*bipartite cores*). Một ví dụ về lõi hai phần là đồ thị hai phần đầy đủ có hướng $K_{3,3}$ với các cạnh có hướng đi từ một phần này sang phần kia, và có thể thêm một số cạnh có hướng vào các phần

Đồ thị web và một số mạng lưới tương tự

Đồ thị web $\mathcal{W}$



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

8

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

Một số tính chất của đồ thị web $\mathcal{W}$ [Zverovich 2021]:

- **Phân phối bậc của $\mathcal{W}$ tuân theo luật lũy thừa với $\gamma = 2.1$** ($\gamma = 2.5$ hoặc 2.7 nếu xét các đồ thị $\mathcal{W}$ có hướng và phân phối bậc ra) và do đó $\mathcal{W}$ cũng được gọi là một **mạng lưới không tỷ lệ (scale-free network)**

- Trong một đồ thị G có n đỉnh, gọi n_k là số đỉnh có bậc k , và dãy $(n_k, 0 \leq k \leq n-1)$ được gọi là **phân phối bậc (degree distribution)** của G
- Phân phối bậc của G được gọi là tuân theo **luật lũy thừa (power law)** nếu với mọi bậc $k > 0$, ta có

$$\frac{n_k}{n} \sim \frac{1}{k^\gamma}, \quad (3)$$

trong đó $\gamma > 1$ là một hằng số cố định

- Số $\mathbb{P}(k) = \frac{n_k}{n}$ là tỷ lệ các đỉnh có bậc k và có thể được xem như xác suất để một đỉnh có bậc k . (Xác suất để một đỉnh có bậc k là tỷ lệ với $k^{-\gamma}$)
- Quan hệ tiệm cận (3) áp dụng cho đồ thị vô hạn, và đối với đồ thị hữu hạn nó có nghĩa là 'xấp xỉ bằng': $\mathbb{P}(k) \approx k^{-\gamma}$. Trong trường hợp G là đồ thị có hướng, luật lũy thừa cho phân phối bậc vào và phân phối bậc ra được định nghĩa tương tự
- Trong các đồ thị khổng lồ tuân theo luật lũy thừa **có các đỉnh với bậc cao**, được gọi là các **trung tâm (hubs)**. Cũng cần lưu ý rằng trong các mạng lưới thực tế, luật lũy thừa có thể có một số sai lệch đối với các đỉnh có bậc rất nhỏ hoặc rất lớn

Đồ thị web và một số mạng lưới tương tự

Đồ thị web $\mathcal{W}$



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

9 Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

Một số tính chất của đồ thị web $\mathcal{W}$ [Zverovich 2021]:

- Đồ thị web $\mathcal{W}$ có *tính chất “thế giới nhỏ” (small-world property)* được giới thiệu trong [Watts and Strogatz 1998]
 - Gọi $d(u, v)$ là khoảng cách giữa các đỉnh u và v trong đồ thị vô hướng G , và gọi X là tập hợp tất cả các cặp đỉnh $\{u, v\}$ phân biệt thỏa mãn $d(u, v)$ hữu hạn. *Khoảng cách trung bình (average distance)* của G là:

$$L(G) = \frac{\sum_{\{u,v\} \in X} d(u, v)}{|X|}$$

- Một tham số tương tự, $L_d(G)$, có thể được định nghĩa cho đồ thị có hướng G , trong đó khoảng cách giữa hai đỉnh là số cạnh trong đường đi ngắn nhất có hướng giữa hai đỉnh đó
- *Tính chất “thế giới nhỏ” (small-world property)* nói rằng $L(G)$ (hoặc $L_d(G)$) *phải nhỏ hơn nhiều so với số đỉnh trong G :* $L(G) \leq c \ln \ln n$, trong đó c là một hằng số nhỏ. Đối với đồ thị web $\mathcal{W}$, các nghiên cứu [Albert, Jeong, and Barabási 1999]; [Broder et al. 2000] cho thấy $L(\mathcal{W}) = 6.8$, trong khi $L_d(\mathcal{W}) = 16$ hoặc 19

Đồ thị web và một số mạng lưới tương tự

Một số mạng lưới tương tự



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

10

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

- **Đồ thị internet (internet graph)** $\mathcal{I}$ có thể được định nghĩa theo nhiều cách khác nhau
 - Ví dụ, các đỉnh của $\mathcal{I}$ có thể đại diện cho các miền (domain) và các cạnh là các kết nối giữa các miền
 - Ngoài ra, các đỉnh của $\mathcal{I}$ có thể là các bộ định tuyến (router) và các cạnh là các kết nối giữa các bộ định tuyến
 - Cả hai loại đồ thị internet này đều tuân theo luật lũy thừa với số mũ $\gamma = 2.5$
- **Đồ thị blog (blog graph)** là một đồ thị con cảm sinh của đồ thị web, trong đó các đỉnh đại diện cho các blog web và các cạnh có hướng là các liên kết giữa các blog. Đồ thị này tuân theo luật lũy thừa với số mũ $\gamma = 2.1$ cho cả phân phối bậc vào và bậc ra, đồng thời có cấu trúc cộng đồng phong phú
- Trong **đồ thị cuộc gọi (call graph)**, các đỉnh đại diện cho các số điện thoại và hai đỉnh được nối với nhau nếu có cuộc gọi từ số này đến số kia trong, chẳng hạn, một ngày. Đồ thị cuộc gọi có hướng cũng tuân theo luật lũy thừa với số mũ $\gamma = 2.1$
- Tìm hiểu thêm về các mạng lưới này trong [Dorogovtsev and Mendes 2003]



- Một trong những mạng xã hội đầu tiên là *đồ thị quen biết (acquaintance graph)*, trong đó mỗi cá nhân được biểu diễn bằng một đỉnh và hai đỉnh được nối với nhau bởi một cạnh nếu các cá nhân tương ứng quen biết nhau.

- *Đường kính (diameter)* của đồ thị G được định nghĩa như sau: $\text{diam}(G) = \max_{u,v \in V} d(u,v)$. Milgram [Milgram 1967]

ngiên cứu đồ thị quen biết và kết luận rằng đường kính của nó khoảng sáu, điều này dẫn đến các câu nói quen thuộc *“sáu chặng phân cách” (six degrees of separation)* và *“Trái Đất tròn” (it's a small world)*



- Trong *bài toán quen biết (acquaintance puzzle)* nổi tiếng, cần phải chứng minh rằng trong một nhóm gồm sáu người, sẽ có ba người hoặc là cùng quen biết nhau hoặc là cùng không quen biết nhau. Bài toán quen biết có liên quan chặt chẽ đến các *số Ramsey (Ramsey number)*—khái niệm quan trọng trong *lý thuyết đồ thị cực hạn (extremal graph theory)* (lý thuyết nghiên cứu sự ảnh hưởng của các tính chất tổng quát trong một đồ thị đến các cấu trúc con cục bộ).

- *Số Ramsey (Ramsey number)* $R_{l,m}$ là số nhỏ nhất sao cho mỗi đồ thị với $R_{l,m}$ đỉnh chứa một đồ thị đầy đủ K_l (trong đó mọi cặp đỉnh nối với nhau) hoặc một tập độc lập (tập hợp các đỉnh mà không có cặp đỉnh nào nối với nhau) m đỉnh
- *Định lý Ramsey (Ramsey's theorem)* nổi tiếng khẳng định các số Ramsey tồn tại, cụ thể là tồn tại một số nguyên dương nhỏ nhất $R_{l,m}$ sao cho mọi tô màu cạnh xanh-đỏ của đồ thị đầy đủ trên $R_{l,m}$ đỉnh đều chứa một đồ thị đầy đủ K_l màu xanh hoặc một đồ thị đầy đủ K_m màu đỏ.



- Trong *đồ thị hợp tác (collaboration graph)*, các đỉnh đại diện cho các nhà nghiên cứu trong một lĩnh vực nào đó và các cạnh nối các nhà nghiên cứu cộng tác với nhau
- Đồ thị hợp tác của các nhà toán học được biết đến như là *đồ thị hợp tác Erdős* (Erdős collaboration graph), trong đó các đỉnh đại diện cho các nhà toán học và hai đỉnh được nối với nhau nếu hai nhà toán học là đồng tác giả của một bài báo chung (bài báo có thể có các đồng tác giả khác)
- Khoảng cách giữa hai nhà toán học trong đồ thị hợp tác Erdős được gọi là *khoảng cách cộng tác (collaboration distance)*. *Số Erdős (Erdős number)* của một nhà toán học là khoảng cách cộng tác của nhà toán học đó với nhà toán học nổi tiếng Paul Erdős (Erdős (1913–1996) có khoảng 500 đồng tác giả và đã xuất bản ít nhất 1525 bài báo, nhiều hơn bất kỳ nhà toán học nào khác trong lịch sử)
- Newman [Newman 2001] nghiên cứu đồ thị hợp tác cho các nhóm khoa học khác nhau và kết luận rằng một đồ thị hợp tác điển hình F có các tính chất sau: $diam(F) \approx 20$ và $L(F) \approx 6$



- Một số mạng xã hội được mô hình hóa bằng đồ thị có hướng nếu mỗi quan hệ giữa các đối tượng thể hiện theo một thứ tự nhất định nào đó; chẳng hạn, nếu một người thấp hơn người khác. Một ví dụ khác là **đồ thị có hướng của các trích dẫn (collaboration graph)** cho tất cả các ấn phẩm trong một lĩnh vực nào đó.
- Một ví dụ khác là **mô hình hóa quan hệ họ hàng** trong một nhóm người: các đỉnh của G là các cá nhân từ nhóm và hai cá nhân x và y được nối với nhau bởi một cạnh có hướng xy nếu x là cha/mẹ của y .
 - Nếu A là ma trận kề của G , thì $A^2 = A \times A$ là ma trận kề của một đồ thị có hướng H . Có thể chứng minh về mặt toán học rằng H biểu diễn mối quan hệ ông bà–cháu trong nhóm người ban đầu. Lưu ý rằng H có thể là một đa đồ thị có hướng nếu không cấm quan hệ cận huyết
 - Ví dụ này có thể được tổng quát hóa hơn nữa cho ma trận kề A^k của đồ thị biểu diễn mối quan hệ tổ tiên qua k thế hệ
- Phân phối bậc của các mạng xã hội **thường tuân theo luật lũy thừa**



- Một *mạng sinh học phân tử (biomolecular network)* bao gồm các đỉnh, đại diện cho các phân tử sinh học (biomolecule), và các cạnh thể hiện sự tương tác giữa chúng
- Các phân tử sinh học có thể là các phân tử lớn (large molecule) (ví dụ: protein, gen) hoặc các phân tử nhỏ (small molecule) (ví dụ: axit nucleic, đường). Chẳng hạn, trong mạng tương tác protein, các cạnh đại diện cho các tiếp xúc vật lý giữa các protein trong một tế bào
- Tương tự như đồ thị web, nhiều mạng tương tác gen, trao đổi chất và protein đều có tính chất “thế giới nhỏ”. Nghiên cứu cho thấy đối với một số mẫu, khoảng cách trung bình nằm trong khoảng từ 3 đến 5, và đường kính nhỏ.



- Một số đỉnh trong một mạng lưới quan trọng hơn những đỉnh khác; chẳng hạn, các protein cụ thể trong mạng lưới tương tác có thể có ý nghĩa cấu trúc quan trọng hơn. Các **các độ đo trung tâm (centrality measures)** đóng vai trò quan trọng trong việc xác định các đỉnh có “thứ hạng cao”.
 - Phân phối bậc của nhiều mạng lưới sinh học đã được chứng minh là tuân theo luật lũy thừa, ví dụ như mạng lưới trao đổi chất và mạng lưới tương tác protein
 - Mạng lưới điều hòa phiên mã (transcriptional regulatory network), là một tập con của mạng lưới điều hòa gen (gene regulatory network), có phân phối bậc ra tuân theo luật lũy thừa, tuy nhiên, phân phối bậc vào tuân theo phân phối hàm mũ
 - Có một số bằng chứng cho thấy bậc của một đỉnh trong mạng lưới tương tác protein chỉ ra khả năng một protein là thiết yếu: các protein không thiết yếu thường có bậc thấp hơn so với bậc của các protein thiết yếu



- Đối với đồ thị liên thông G , **độ lệch tâm (eccentricity)** của một đỉnh u được định nghĩa như sau: $e(u) = \max_{v \in V} d(u, v)$.

Khi đó, **tâm (centre)** của G là tập hợp

$\{u \in V : e(u) = \min_{v \in V} e(v)\}$. Nói cách khác, tâm của đồ thị

bao gồm các đỉnh có độ lệch tâm nhỏ nhất.

- Trong mạng lưới trao đổi chất, tâm có thể đại diện cho một số thành phần chính của quá trình trao đổi chất của sinh vật và có vai trò như một “nút thắt cổ chai”



- **Độ trung tâm trung gian (betweenness centrality)** là một độ đo khác để định lượng tầm quan trọng của một đỉnh—đỉnh quan trọng thường thuộc về nhiều đường đi ngắn nhất giữa các đỉnh khác, từ đó kiểm soát “sự liên lạc (communication)” giữa các đỉnh này

- Gọi σ_{vw} là số đường đi ngắn nhất giữa các đỉnh v và w trong đồ thị G , và $\sigma_{vw}(u)$ là số đường đi ngắn nhất (v, w) đi qua đỉnh u , trong đó u, v và w đều khác nhau.
- Độ trung tâm trung gian của u được tính bởi công thức:

$$B(u) = \sum_{v, w \in V, v \neq u \neq w} \frac{\sigma_{vw}(u)}{\sigma_{vw}}$$

- Ví dụ, trong mạng lưới tương tác protein của men nở (yeast protein interaction network), độ trung tâm trung gian trung bình cho các protein thiết yếu cao hơn nhiều so với các protein không thiết yếu

Thuật toán Gale-Shapley

Giới thiệu



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

- Bài toán được đặt ra: Ghép cặp giữa các thực tập sinh y khoa và các bệnh viện
- Mỗi bệnh viện có một danh sách ưu tiên về các thực tập sinh
- Mỗi thực tập sinh có một danh sách ưu tiên về các bệnh viện
- Mục tiêu: Tìm một cách ghép cặp sao cho không tồn tại cặp (*bệnh viện*, *thực tập sinh*) nào đều thích nhau hơn đối tượng hiện tại của họ
- Thuật toán Gale-Shapley (1962) giải quyết bài toán này [Gale and Shapley 1962]

19

28

Thuật toán Gale-Shapley

Định nghĩa và Mô tả thuật toán



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

Ghép cặp ổn định (Stable Matching)

Cho hai tập hợp các đối tượng A và B có cùng kích thước. Mỗi đối tượng trong A và B có danh sách xếp hạng ưu tiên riêng về tất cả các đối tượng trong tập kia. Một ghép cặp M là **ổn định (stable)** nếu không tồn tại cặp (a, b) với $a \in A$ và $b \in B$ sao cho:

- (1) a và b không được ghép với nhau trong M
- (2) a thích b hơn đối tượng hiện tại của a trong M
- (3) b thích a hơn đối tượng hiện tại của b trong M

■ Ở mỗi vòng lặp của thuật toán

- Mỗi bệnh viện chưa ghép đôi gửi đề nghị đến thực tập sinh cao nhất trong danh sách mà họ chưa gửi đề nghị trước đó
 - Mỗi thực tập sinh chọn đề nghị tốt nhất trong số các đề nghị nhận được và từ chối các đề nghị khác
 - Bệnh viện bị từ chối sẽ gửi đề nghị đến ứng viên tiếp theo trong danh sách của họ trong vòng kế tiếp
- Thuật toán kết thúc khi mọi bệnh viện đều được ghép đôi hoặc đã gửi đề nghị đến tất cả ứng viên trong danh sách của họ

20

28

Thuật toán Gale-Shapley

Định nghĩa và Mô tả thuật toán



Thuật toán 1: Thuật toán Gale-Shapley (bệnh viện đề xuất)

Input: Danh sách ưu tiên của bệnh viện và thực tập sinh

Output: Tập các cặp ghép đôi ổn định

- 1 Khởi tạo tất cả bệnh viện và thực tập sinh đều chưa được ghép đôi
- 2 **while** có bệnh viện h chưa được ghép đôi và h chưa gửi đề nghị đến tất cả thực tập sinh **do**
 - 3 $s \leftarrow$ thực tập sinh cao nhất trong danh sách ưu tiên của h mà h chưa gửi đề nghị
 - 4 **if** s chưa được ghép đôi **then**
 - 5 $\quad$ ghép đôi h và s
 - 6 **else**
 - 7 $h' \leftarrow$ bệnh viện hiện tại của s
 - 8 **if** s thích h hơn h' **then**
 - 9 $\quad$ hủy ghép đôi (h', s)
 - 10 $\quad$ ghép đôi (h, s)
 - 11 $\quad$ đặt h' là chưa được ghép đôi
- 12 **return** tập các cặp ghép đôi

Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

21 Thuật toán Gale-Shapley

Tài liệu tham khảo

Thuật toán Gale-Shapley

Ví dụ minh họa



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

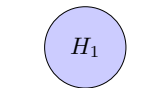
Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo



$S_1 > S_2 > S_3$



$S_2 > S_1 > S_3$



$S_1 > S_3 > S_2$



$H_2 > H_1 > H_3$



$H_1 > H_2 > H_3$



$H_1 > H_2 > H_3$

Khởi tạo: Mọi bệnh viện (H_i) và thực tập sinh (S_i) đều chưa ghép đôi

22

28

Thuật toán Gale-Shapley

Ví dụ minh họa



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

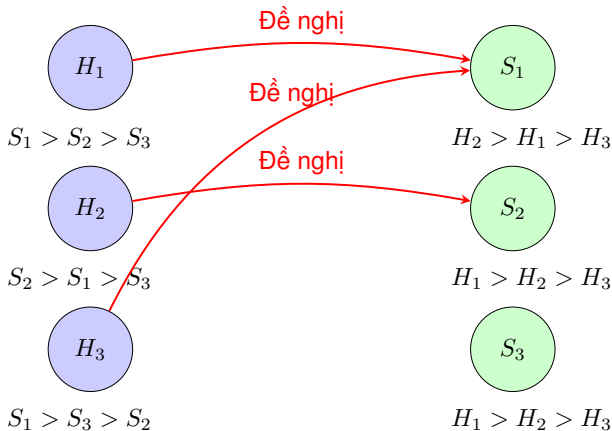
Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo



Vòng 1: Mỗi bệnh viện gửi đề nghị đến thực tập sinh ưa thích nhất

Thuật toán Gale-Shapley

Ví dụ minh họa



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

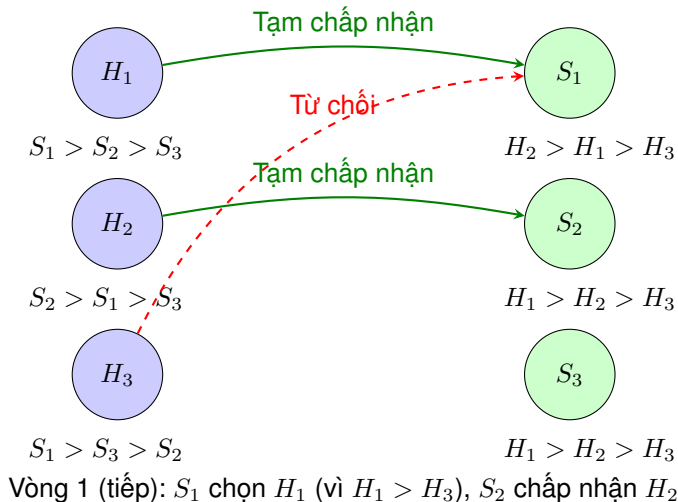
Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

22



28

Thuật toán Gale-Shapley

Ví dụ minh họa



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

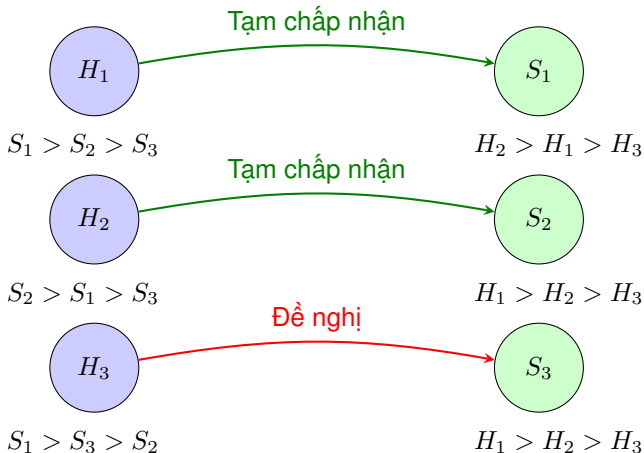
Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

22



Vòng 2: H_3 bị từ chối, gửi đề nghị đến thực tập sinh thứ 2 là S_3

28

Thuật toán Gale-Shapley

Ví dụ minh họa



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

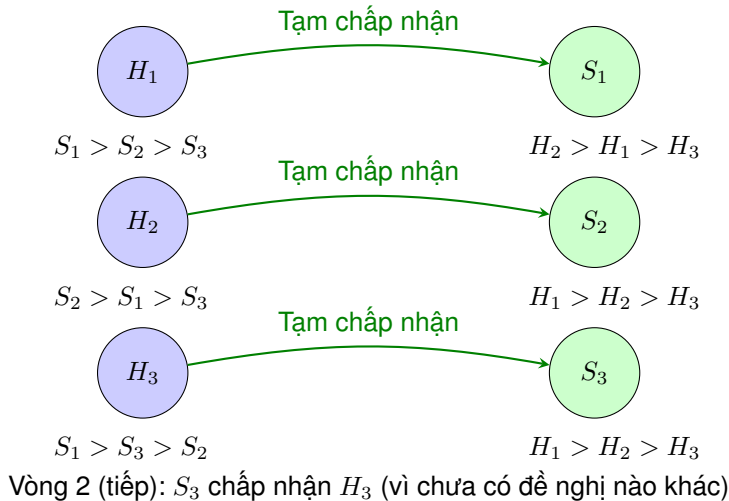
Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

22



Thuật toán Gale-Shapley

Ví dụ minh họa



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

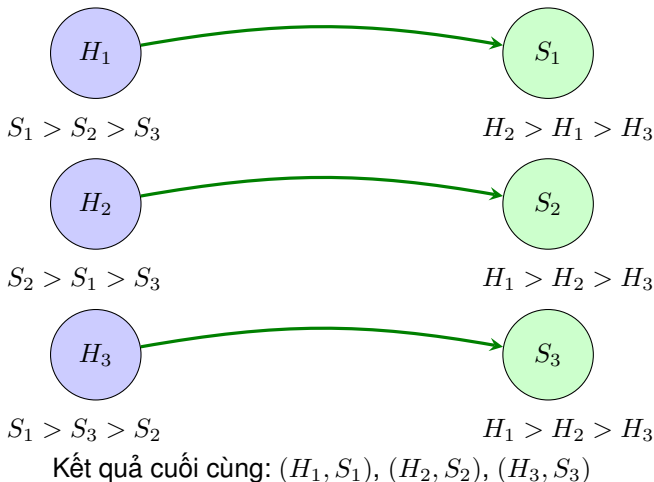
Mạng xã hội

Mạng sinh học

22

Thuật toán Gale-Shapley

Tài liệu tham khảo



Thuật toán Gale-Shapley

Một ví dụ phức tạp hơn



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

23

Thuật toán Gale-Shapley

Tài liệu tham khảo



$S_1 > S_2 > S_3 > S_4$



$S_2 > S_1 > S_4 > S_3$



$S_3 > S_1 > S_2 > S_4$



$S_4 > S_3 > S_1 > S_2$



$H_2 > H_1 > H_3 > H_4$



$H_1 > H_3 > H_4 > H_2$



$H_4 > H_2 > H_3 > H_1$



$H_3 > H_1 > H_2 > H_4$

Khởi tạo: Danh sách ưu tiên của các bệnh viện (H_i) và thực tập sinh (S_i)

28

Thuật toán Gale-Shapley

Một ví dụ phức tạp hơn



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

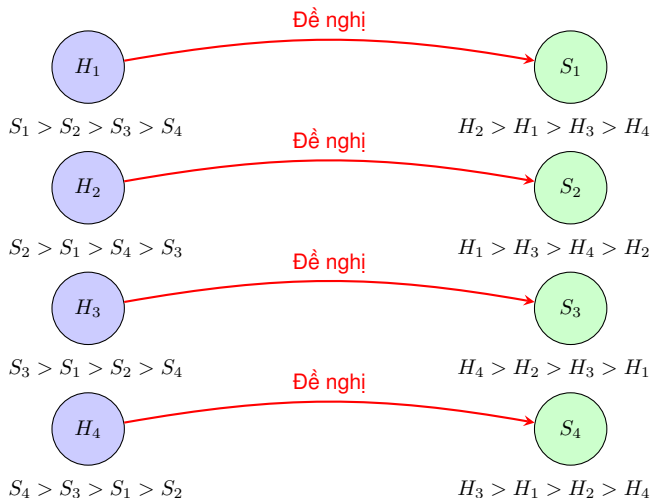
Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

23

28



Vòng 1: Mỗi bệnh viện gửi đề nghị đến thực tập sinh ưa thích nhất

Thuật toán Gale-Shapley

Một ví dụ phức tạp hơn



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

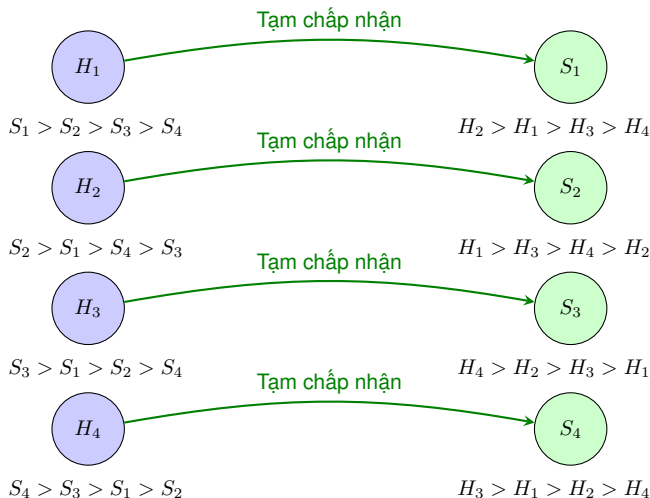
Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

23

28



Vòng 1 (tiếp): Tất cả thực tập sinh đều chấp nhận đề nghị

Thuật toán Gale-Shapley

Một ví dụ phức tạp hơn



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

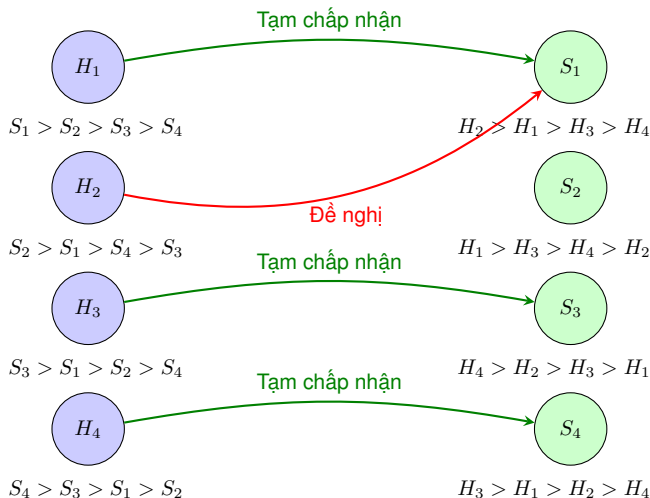
Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

23



28

Thuật toán Gale-Shapley

Một ví dụ phức tạp hơn



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

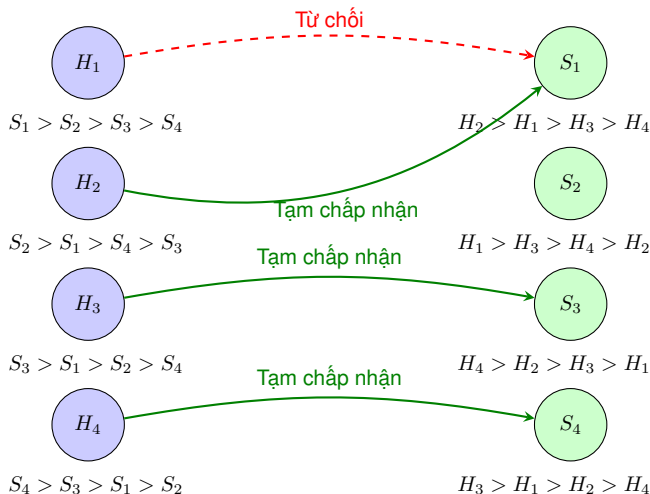
Mạng xã hội

Mạng sinh học

23

Thuật toán Gale-Shapley

Tài liệu tham khảo



Vòng 2 (tiếp): S_1 thích H_2 hơn H_1 nên chấp nhận H_2 và từ chối H_1

28

Thuật toán Gale-Shapley

Một ví dụ phức tạp hơn



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

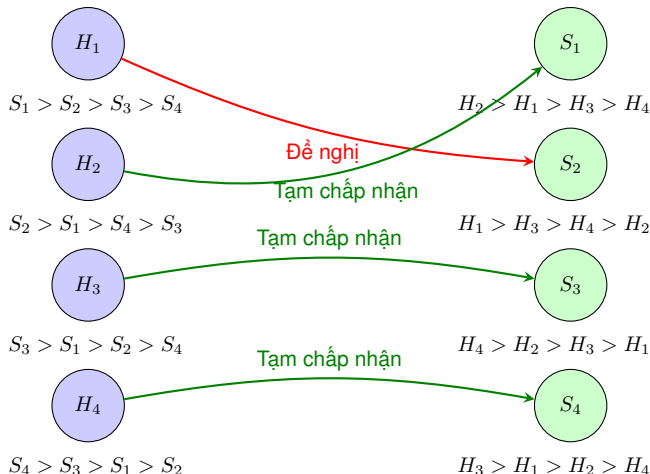
Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

23

28



Vòng 3: H_1 bị từ chối, nên đề nghị với thực tập sinh thứ 2 là S_2

Thuật toán Gale-Shapley

Một ví dụ phức tạp hơn



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

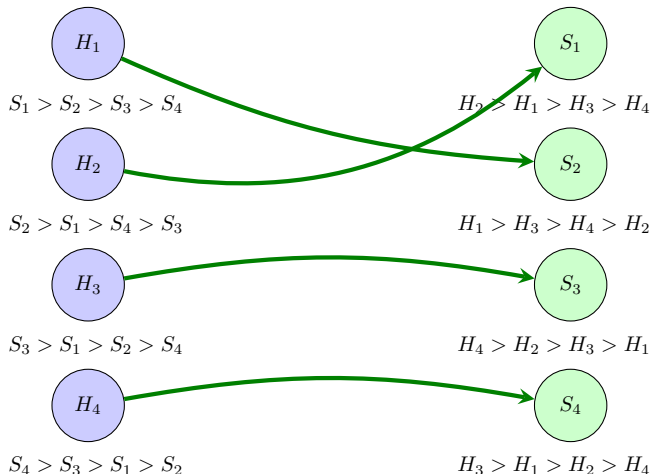
Mạng sinh học

23

Thuật toán Gale-Shapley

Tài liệu tham khảo

28



Kết quả cuối: $(H_1, S_2), (H_2, S_1), (H_3, S_3), (H_4, S_4)$

Thuật toán Gale-Shapley

Tính chất



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

Tài liệu tham khảo

Định lý 8: Tính chất của thuật toán Gale-Shapley

1. Thuật toán luôn kết thúc sau hữu hạn bước
2. Kết quả là một cặp ghép ổn định
3. Kết quả là cặp ghép ổn định tối ưu cho bên đề xuất (bệnh viện)
4. Kết quả là cặp ghép ổn định tệ nhất cho bên được đề xuất (thực tập sinh)

- Thuật toán Gale-Shapley có độ phức tạp thời gian $O(n^2)$, trong đó n là số lượng thành phần trong mỗi tập hợp
- Được áp dụng rộng rãi trong nhiều lĩnh vực: ghép cặp sinh viên với trường đại học, ghép cặp thực tập sinh y khoa với bệnh viện, ...

24

28

Tài liệu tham khảo



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

 Zverovich, Vadim (2021). *Modern Applications of Graph Theory*. Oxford University Press. ISBN:

978-0-19-885674-0. DOI:

10.1093/oso/9780198856740.001.0001.

 Babai, László (2016). “Graph Isomorphism in Quasipolynomial Time”. In: *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, pp. 684–697. DOI:

10.1145/2897518.2897542.

 Rosen, Kenneth (2012). *Discrete Mathematics and Its Applications*. 7th. McGraw-Hill.

 Baldi, Pierre, Paolo Frasca, and Padhraic Smyth (2003). *Modeling the Internet and the Web, Probabilistic Methods and Algorithms*. John Wiley & Sons. ISBN:

978-0-470-84906-4. DOI: 10.1002/047086844X.

25

Tài liệu tham khảo

28

Tài liệu tham khảo (tiếp)



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

 Dorogovtsev, S. N. and J. F. F. Mendes (2003). *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford: Oxford University Press. ISBN: 9780198515906. DOI: 10.1093/acprof:oso/9780198515906.001.0001.

 Newman, Mark E. J. (2001). "The structure of scientific collaboration networks". In: *Proceedings of the National Academy of Sciences of the USA* 98.2, pp. 404–409. DOI: 10.1073/pnas.98.2.404.

 Broder, Andrei, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener (2000). "Graph structure in the web". In: *Computer Networks* 33.1-6, pp. 309–320. DOI: 10.1016/S1389-1286(00)00083-9.

26

Tài liệu tham khảo

28

Tài liệu tham khảo (tiếp)



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley

27

Tài liệu tham khảo



Albert, Réka, Hawoong Jeong, and Albert-László Barabási (1999). “Diameter of the world-wide web”. In: *Nature* 401.6749, pp. 130–131. DOI: 10.1038/43601.



Watts, Duncan J. and Steven H. Strogatz (1998). “Collective dynamics of ‘small-world’ networks”. In: *Nature* 393.6684, pp. 440–442. DOI: 10.1038/30918.



Harary, Frank and Ronald C. Read (1973). “Is the null graph a pointless concept?” In: *Graphs and Combinatorics*. Proceedings of the Conference at George Washington University. New York, NY: Springer-Verlag.



Hopcroft, John E. and Richard M. Karp (1973). “An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs”. In: *SIAM Journal on Computing* 2.4, pp. 225–231. DOI: 10.1137/0202019.

28

Tài liệu tham khảo (tiếp)



Lý thuyết đồ thị I

Hoàng Anh Đức

Một số lỗi thường gặp

Một số ứng dụng của đồ thị

Đồ thị web và một số mạng lưới tương tự

Mạng xã hội

Mạng sinh học

Thuật toán Gale-Shapley



Milgram, Stanley (1967). “The small world problem”. In: *Psychology Today* 2.1, pp. 60–67.



Edmonds, Jack (1965). “Paths, trees, and flowers”. In: *Canadian Journal of Mathematics* 17, pp. 449–467. DOI: 10.4153/CJM-1965-045-4.



Gale, David and Lloyd S. Shapley (1962). “College admissions and the stability of marriage”. In: *The American Mathematical Monthly* 69.1, pp. 9–14. DOI: 10.2307/2312726.



Ford, L. R. and D. R. Fulkerson (1956). “Maximal flow through a network”. In: *Canadian Journal of Mathematics* 8, pp. 399–404. DOI: 10.4153/CJM-1956-045-5.



Kuhn, Harold W. (1955). “The Hungarian method for the assignment problem”. In: *Naval Research Logistics Quarterly* 2.1-2, pp. 83–97. DOI: 10.1002/nav.3800020109.

1. Logic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Đệ quy
4. Thuật toán I: Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp
5. Thuật toán II: Thuật toán tham lam, thuật toán đệ quy
6. Lý thuyết số cơ bản
7. Các phương pháp đếm
8. Lý thuyết đồ thị I: Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông
9. Lý thuyết đồ thị II: Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị
10. Lý thuyết đồ thị III: Cây
11. Đại số Boole

VNU-HUS MAT3500: Toán rời rạc

Lý thuyết đồ thị II

Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn



Nội dung



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

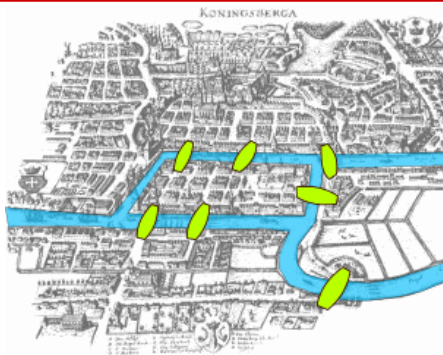
Tô màu đồ thị phẳng

Đường đi Euler và Đường đi Hamilton

Bảy cây cầu ở Königsberg



Hình: Leonhard Euler
1707–1783 (Wikipedia)



Hình: Bản đồ Königsberg cũ (Wikipedia)

Bảy cây cầu ở Königsberg

Tìm một tuyến đường đi qua mỗi cây cầu chính xác một lần và quay lại vị trí xuất phát

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

2

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Đường đi Euler và Đường đi Hamilton

Bảy cây cầu ở Königsberg



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

3

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

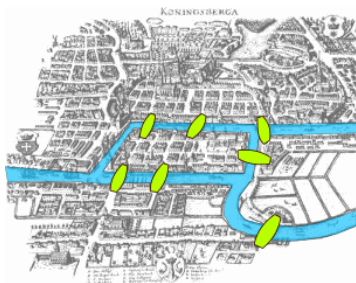
Định lý Kuratowski

Tô màu đồ thị

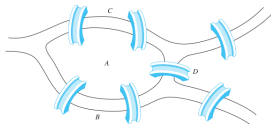
Giới thiệu

Một số tính chất cơ bản

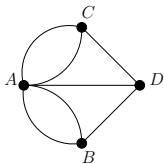
Tô màu đồ thị phẳng



(a) Bản đồ Königsberg cũ (Wikipedia)



(b) Bản đồ Königsberg cũ đơn giản hóa



(c) Đồ thị tương ứng

■ Đồ thị tương ứng:

- Mỗi vùng đất ứng với một đỉnh
- Mỗi cây cầu nối hai vùng đất ứng với một cạnh

■ Tìm chu trình đơn trong đồ thị chứa tất cả các cạnh

Đường đi Euler và Đường đi Hamilton

Đường đi Euler

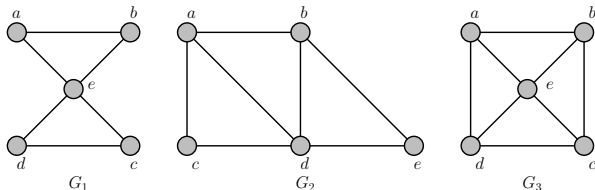


Lý thuyết đồ thị II

Hoàng Anh Đức

Cho đồ thị $G = (V, E)$. Một **đường đi/chu trình Euler (Eulerian path/circuit)** trong G là một đường đi/chu trình đơn có chứa mọi cạnh của G

Ví dụ 1



- G_1 có chu trình Euler, G_2 và G_3 không có
- G_2 có đường đi Euler, G_3 không có

Bài tập 1

Chứng minh rằng nếu $G = (V, E)$ là một đa đồ thị vô hướng thỏa mãn $\deg_G(u) \geq 2$ với mọi $u \in V$ thì G có một chu trình đơn

Đường đi Euler và
Đường đi Hamilton

4 Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Đường đi Euler và Đường đi Hamilton

Đường đi Euler



Lý thuyết đồ thị II

Hoàng Anh Đức

Định lý 1: [Euler 1736]; [Hierholzer and Wiener 1873]

Một đa đồ thị vô hướng liên thông có một chu trình Euler khi và chỉ khi mỗi đỉnh của đồ thị có bậc chẵn

Chứng minh.

($\Rightarrow$) Giả sử một đa đồ thị vô hướng liên thông $G = (V, E)$ có một chu trình Euler $e_1, e_2, \dots, e_m$ trong đó $e_i = x_{i-1}x_i \in E$ với $1 \leq i \leq m$ và $x_0 = x_m = u$.

- Với $v = x_i$ ($2 \leq i \leq m-1$): chu trình đi vào v qua e_i và đi ra qua e_{i+1}
- Với $u = x_0 = x_m$: chu trình đi ra u qua e_1 và trở lại u qua e_m



5

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Đường đi Euler và Đường đi Hamilton

Đường đi Euler



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

6

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

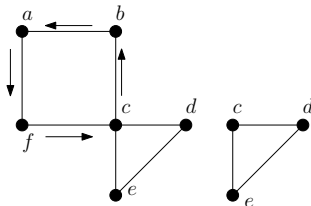
Tô màu đồ thị phẳng

Chứng minh (tiếp).

($\Leftarrow$) Giả sử mọi đỉnh của G đều có bậc chẵn. Lặp lại quá trình chọn chu trình sau cho đến khi đã chọn hết các cạnh
(Thuật toán Hierholzer (1873))

- Xuất phát từ đỉnh $x_0 = a$ bất kỳ
- Xây dựng một đường đi đơn bằng cách chọn tùy ý các cạnh $x_0x_1, x_1x_2, \dots, x_{k-1}x_k$ để thêm vào đường đi cho đến khi không chọn được nữa
- Do bậc của mỗi đỉnh là chẵn, với mỗi đỉnh x_i , ta luôn có thể đi vào từ cạnh $x_{i-1}x_i$ và đi ra từ cạnh x_ix_{i+1} . Do bậc của a cũng phải là chẵn, cạnh cuối cùng được chọn sẽ có dạng ya
- Bỏ đi các cạnh đã chọn và các đỉnh không kề với các cạnh còn lại

Cuối cùng, ghép các chu trình trên thành một chu trình Euler.
(Thuật toán chạy trong thời gian $O(|E|)$)



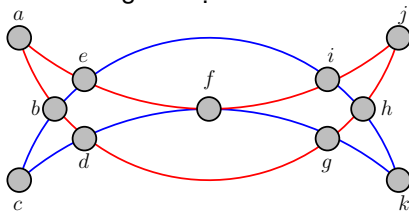
Đường đi Euler và Đường đi Hamilton

Đường đi Euler



Ví dụ 2

Tìm chu trình Euler trong đồ thị sau



■ Bắt đầu từ $x_0 = a$, chọn tùy ý các cạnh $x_0x_1, x_1x_2, \dots, x_{k-1}x_k, x_k a$. Ví dụ: **ae, ef, fi, ij, jh, hg, gd, db, ba**

■ Bỏ đi các cạnh đã chọn và các đỉnh a, j

■ Bắt đầu từ $x_0 = c$, chọn tùy ý các cạnh $x_0x_1, x_1x_2, \dots, x_{l-1}x_l, x_l c$. Ví dụ: **cb, be, ei, ih, hk, kg, gf, fd, dc**

■ Bỏ đi các cạnh đã chọn và các đỉnh cô lập còn lại

■ Ghép hai chu trình đã chọn thành một chu trình Euler:

■ **ae, ei, ih, hk, kg, gf, fd, dc, cb, be, ef, fi, ij, jh, hg, gd, db, ba**

■ **ae, ef, fd, dc, cb, be, ei, ih, hk, kg, gf, fi, ij, jh, hg, gd, db, ba**

■ ...

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

7

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Đường đi Euler



Với những giá trị nào của n thì các đồ thị sau có chu trình Euler?

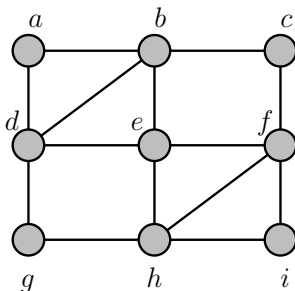
(a) K_n

(c) W_n

(b) C_n

(d) Q_n

Tìm chu trình Euler trong đồ thị sau



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và Đường đi Hamilton

8

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi ngắn nhất

Đỗ thi có trong số

Thuật toán Dijkstra

Đỏ thi phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thi

Giới thiệu

Một số tính chất cơ bản

Tổ màu đỏ thi phẳng

Đường đi Euler và Đường đi Hamilton

Đường đi Euler



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Định lý 2

Một đa đồ thị vô hướng liên thông G có đường đi Euler nhưng không có chu trình Euler khi và chỉ khi có đúng hai đỉnh của G có bậc lẻ.

Chứng minh.

($\Rightarrow$) Giả sử G có đường đi Euler nhưng không có chu trình Euler

- Hai đỉnh ở hai đầu mút của đường đi có bậc lẻ
- Các đỉnh còn lại có bậc chẵn

($\Leftarrow$) Giả sử G có chính xác hai đỉnh bậc lẻ u, v

- Tìm chu trình Euler của đồ thị $G + uv$
- Xóa cạnh uv trong chu trình để thu được đường đi Euler trong G



Đường đi Euler và Đường đi Hamilton

Đường đi Euler

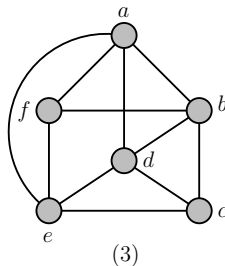
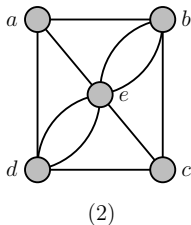
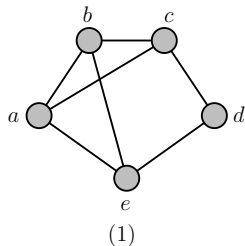


Lý thuyết đồ thị II

Hoàng Anh Đức

Bài tập 4

Hãy xác định xem các đồ thị sau có chu trình/đường đi Euler hay không? Nếu có, hãy tìm một chu trình/đường đi Euler trong đồ thị đó



10

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

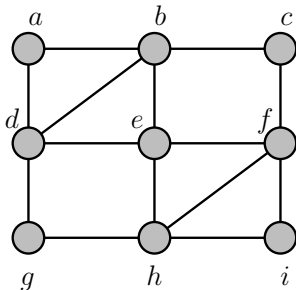
Đường đi Euler và Đường đi Hamilton

Đường đi Euler



Bài tập 5

Thuật toán Fleury [Fleury 1883] xây dựng các chu trình Euler bằng cách đầu tiên chọn một đỉnh tùy ý của một đa đồ thị liên thông, và sau đó tạo thành một chu trình bằng cách chọn các cạnh liên tiếp. Một khi một cạnh được chọn, nó sẽ bị loại bỏ. Các cạnh được chọn liên tiếp sao cho mỗi cạnh bắt đầu từ nơi cạnh cuối cùng kết thúc, và sao cho cạnh này không phải là một cạnh cắt trừ khi không có lựa chọn thay thế nào khác. Sử dụng thuật toán Fleury để tìm một chu trình Euler trong đồ thị sau



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

11

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Đường đi Euler và Đường đi Hamilton

Đường đi Euler



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

12 Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Tương tự, đối với đồ thị có hướng $G = (V, E)$, một **đường đi/chu trình Euler (Eulerian path/circuit)** trong G là một đường đi/chu trình đơn có hướng chứa mọi cạnh của G

Định lý 3

Một đa đồ thị có hướng liên thông mạnh có chu trình Euler khi và chỉ khi mọi đỉnh của đồ thị có bậc vào bằng bậc ra

- Điều kiện “liên thông mạnh” có thể được thay bằng “liên thông yếu” (**Tại sao?**)

Định lý 4

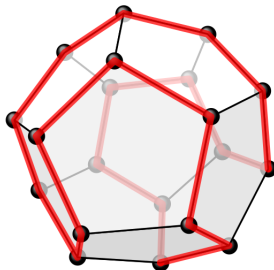
Một đa đồ thị có hướng liên thông yếu có đường đi Euler nhưng không có chu trình Euler khi và chỉ khi có đúng hai đỉnh u, v của G sao cho $\deg^-(u) = \deg^+(u) + 1$ và $\deg^-(v) = \deg^+(v) - 1$

Đường đi Euler và Đường đi Hamilton

Trò chơi “Vòng quanh thế giới” (1857)



Hình: Sir William Rowan Hamilton 1805-1865 (Wikipedia)



Hình: Trò chơi “Vòng quanh thế giới” (Wikipedia)

Trò chơi “Vòng quanh thế giới”

Mỗi đỉnh trong 20 đỉnh của khối 12 mặt đại diện cho một thành phố. Tìm đường đi xuất phát từ một đỉnh dọc theo các cạnh của khối, ghé thăm mỗi đỉnh còn lại một lần, và quay lại vị trí ban đầu

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

13

58

Đường đi Euler và Đường đi Hamilton

Trò chơi “Vòng quanh thế giới” (1857)



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

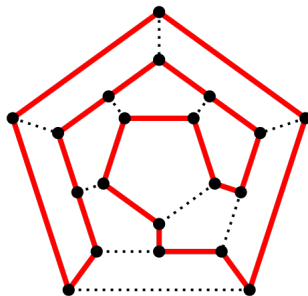
Một số tính chất cơ bản

Tô màu đồ thị phẳng

14



(a) Trò chơi “Vòng quanh thế giới”
(Wikipedia)



(b) Đồ thị đẳng cấu với khối 12 mặt
(Wikipedia)

Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

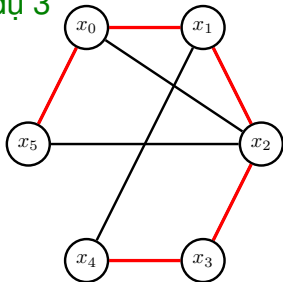
Một số tính chất cơ bản

Tô màu đồ thị phẳng

Cho $G = (V, E)$ là một đồ thị vô hướng gồm $n \geq 1$ đỉnh

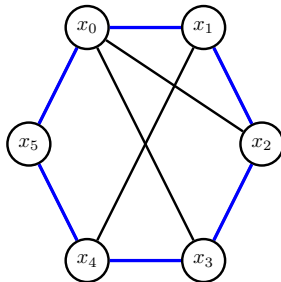
- Một **đường đi Hamilton** trong G là một đường đi đơn $x_0, x_1, \dots, x_{n-1}$ thỏa mãn điều kiện $V = \{x_0, x_1, \dots, x_{n-1}\}$ và $x_i \neq x_j$ với $0 \leq i < j \leq n-1$
- Một **chu trình Hamilton** trong G là một chu trình đơn $x_0, x_1, \dots, x_{n-1}, x_0$ thỏa mãn điều kiện $x_0, x_1, \dots, x_{n-1}$ là một đường đi Hamilton

Ví dụ 3



Đường đi Hamilton

$x_5, x_0, x_1, x_2, x_3, x_4$



Chu trình Hamilton

$x_0, x_1, x_2, x_3, x_4, x_5, x_0$

15

58

Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

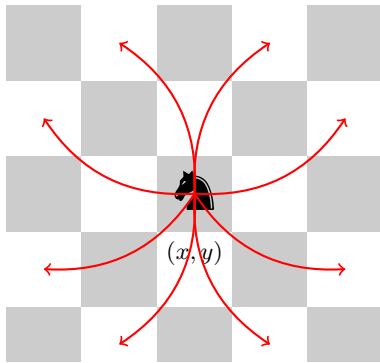
Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Ví dụ 4 (Bài toán quân mã đi tuần (Knight's tour))

Quân mã (knight) là một quân cờ có thể di chuyển hai ô theo chiều ngang và một ô theo chiều dọc, hoặc một ô theo chiều ngang và hai ô theo chiều dọc trên bàn cờ. Nghĩa là, một quân mã tại ô (x, y) có thể di chuyển đến một trong tám ô $(x \pm 2, y \pm 1)$, $(x \pm 1, y \pm 2)$, nếu những ô này nằm trên bàn cờ, như được minh họa dưới đây



16

58

Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Ví dụ 4 (tiếp)

- **Đường đi tuần của quân mã (knight's tour)** là một chuỗi các nước đi hợp lệ của quân mã bắt đầu từ một ô nào đó và thăm mỗi ô đúng một lần $\Leftrightarrow$ **đường đi Hamilton** trong đồ thị tương ứng
- Đường đi tuần được gọi là **khép kín (reentrant)** nếu có một nước đi hợp lệ đưa quân mã từ ô cuối cùng trở lại ô xuất phát $\Leftrightarrow$ **chu trình Hamilton** trong đồ thị tương ứng
- Có thể mô hình hóa bằng cách sử dụng đồ thị:
 - Một đỉnh cho mỗi ô trên bàn cờ
 - Một cạnh nối hai đỉnh nếu quân mã có thể di chuyển hợp lệ giữa hai ô tương ứng với các đỉnh đó

17

58

Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

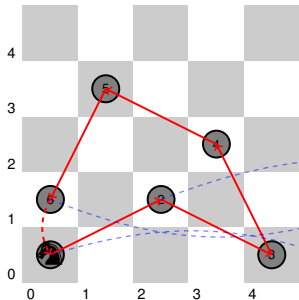
Định lý Kuratowski

Tô màu đồ thị

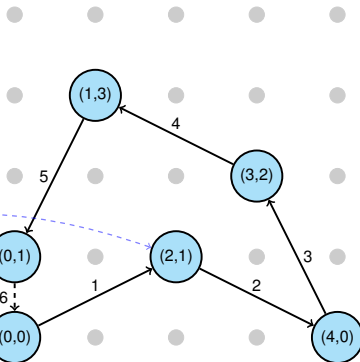
Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng



(a) Bàn cờ 5×5 với một vài bước nhảy của quân mã



(b) Mô hình đồ thị tương ứng

Hình: Một số bước đi của quân mã trên bàn cờ 5×5 và các cạnh tương ứng trên đồ thị được mô tả. Quân mã xuất phát từ góc dưới cùng bên trái (ô $(0,0)$). Mũi tên với nét liền (nét đứt) thể hiện quân mã nhảy (không nhảy được) được từ ô ban đầu đến ô kế tiếp

Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Bài tập 6

- (a) Vẽ đồ thị biểu diễn các nước đi hợp lệ của quân mã trên bàn cờ 3×3
- (b) Vẽ đồ thị biểu diễn các nước đi hợp lệ của quân mã trên bàn cờ 3×4

Bài tập 7

Chứng minh rằng đồ thị biểu diễn các nước đi hợp lệ của quân mã trên bàn cờ kích thước $m \times n$, với mọi số nguyên dương m và n , là đồ thị hai phần

19

58

Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

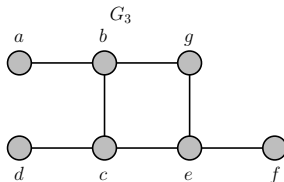
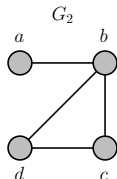
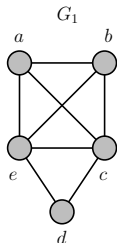
Tô màu đồ thị phẳng

20

- Chưa có một thuật toán trong thời gian đa thức để kiểm tra xem một đồ thị có chu trình Hamilton hay không
- Một số tính chất có thể được sử dụng để chỉ ra một đồ thị không có chu trình Hamilton
 - Đồ thị có chứa đỉnh bậc 1 không có chu trình Hamilton
 - Nếu đỉnh v của đồ thị G có bậc 2 thì hai cạnh kề với v thuộc mọi chu trình Hamilton của G (nếu có)
 - Một chu trình Hamilton không chứa một chu trình con nào có số đỉnh nhỏ hơn nó

Bài tập 8

Các đồ thị sau có chu trình/đường đi Hamilton không?



58

Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Bài tập 9

Hãy cho ví dụ về một đồ thị mà chu trình Euler của nó cũng là chu trình Hamilton

Bài tập 10

Một thông báo chẩn đoán (diagnostic message) có thể được gửi qua mạng máy tính để thực hiện kiểm tra trên tất cả các liên kết và trong tất cả các thiết bị. Loại đường đi nào nên được sử dụng để kiểm tra tất cả các liên kết? Để kiểm tra tất cả các thiết bị?

21

58

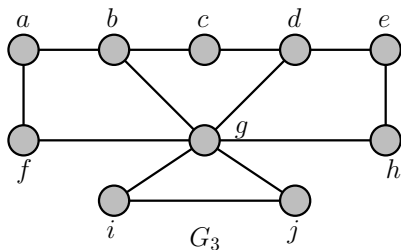
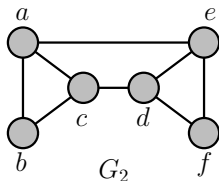
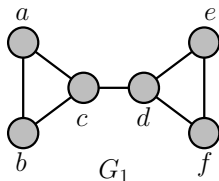
Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



Bài tập 11

Đồ thị nào sau đây có chu trình Hamilton? Tại sao?



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

22

58

Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



Định lý 5: Định lý Dirac

Nếu $G = (V, E)$ là một đơn đồ thị vô hướng gồm n đỉnh ($n \geq 3$) thỏa mãn điều kiện bậc của mỗi đỉnh trong G lớn hơn hoặc bằng $n/2$ thì G có một chu trình Hamilton

Bài tập 12 ((*) Chứng minh Định lý Dirac)

- (a) Để thấy Định lý đúng với $n = 3$. Giả sử $n \geq 4$
- (b) G phải liên thông (Tại sao?)
- (c) Gọi $P = v_0, v_1, \dots, v_k$ là đường đi đơn có độ dài lớn nhất trong G ($0 \leq k \leq n - 1$).
 - Mọi đỉnh kề với v_0 hoặc v_k đều phải thuộc P (Tại sao?)
 - Do $\deg(v_k) \geq n/2$, có ít nhất $n/2$ cạnh phân biệt $v_i v_{i+1}$ của P thỏa mãn $v_i v_k \in E$. Tương tự, do $\deg(v_0) \geq n/2$, có ít nhất $n/2$ cạnh phân biệt $v_j v_{j+1}$ của P thỏa mãn $v_0 v_{j+1} \in E$
 - Do P có ít hơn n cạnh, tồn tại một cạnh $v_q v_{q+1}$ thỏa mãn đồng thời hai điều kiện trên: $v_q v_k \in E$ và $v_0 v_{q+1} \in E$
- (d) P chứa tất cả các đỉnh của G (Tại sao?)

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

23

58

Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



Lý thuyết đồ thị II

Hoàng Anh Đức

Định lý 6: Định lý Ore

Nếu $G = (V, E)$ là một đơn đồ thị vô hướng gồm n đỉnh ($n \geq 3$) thỏa mãn điều kiện $\deg(u) + \deg(v) \geq n$ với mọi cặp đỉnh u, v không kề nhau trong G thì G có một chu trình Hamilton

Bài tập 13

Chứng minh Định lý Dirac (Định lý 5) bằng cách sử dụng Định lý Ore

Bài tập 14 (★)

Cho $G = (V_1 \cup V_2, E)$ là một đồ thị hai phần với $|V_1| = |V_2| = n$ ($n \geq 2$). Chứng minh rằng nếu $\deg(v) > n/2$ với mọi đỉnh $v \in V = V_1 \cup V_2$ thì G có một chu trình Hamilton (**Gợi ý:** Xem lại chứng minh của Định lý Dirac.)

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

24

58

Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

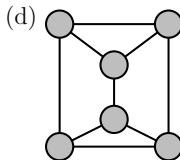
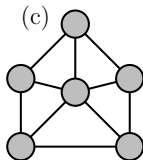
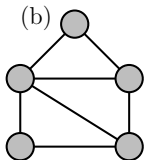
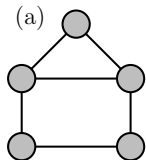
Một số tính chất cơ bản

Tô màu đồ thị phẳng

Bài tập 15

Với mỗi đồ thị sau, hãy xác định

- (i) có thể sử dụng Định lý Dirac để chứng minh đồ thị có chu trình Hamilton không?
- (ii) có thể sử dụng Định lý Ore để chứng minh đồ thị có chu trình Hamilton không?
- (iii) đồ thị có chu trình Hamilton không?



25

58

Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

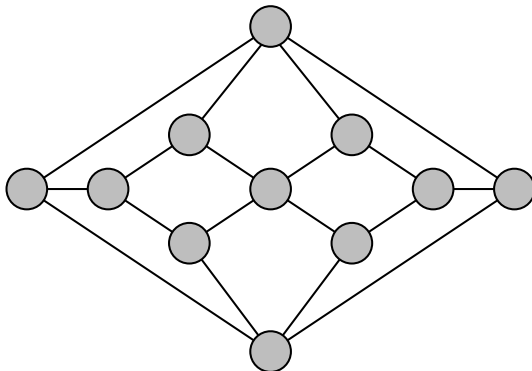
Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Bài tập 16

Chứng minh rằng đồ thị sau đây không có chu trình Hamilton.
(**Gợi ý:** Đồ thị đã cho là một đồ thị hai phần)



26

58

Bài toán đường đi ngắn nhất

Đồ thị có trọng số



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton
Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

27

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

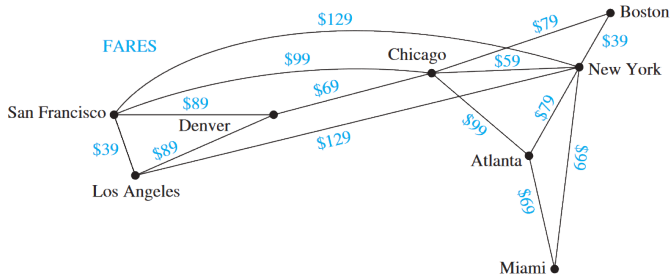
Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng



Hình: Đồ thị có trọng số mô tả giá vé của các chuyến bay giữa một số thành phố ở Mỹ (từ [Rosen 2012], Chương 10)

Bài toán đường đi ngắn nhất

Đồ thị có trọng số



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

28

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

Cho $G = (V, E, w)$ là đơn đồ thị có trọng số

- Một đường đi từ u đến v qua các cạnh (cung) $e_1, e_2, \dots, e_n$

có **độ dài (length)** $c(u, v) = \sum_{i=1}^n w(e_i)$

- **Khoảng cách (distance)** giữa hai đỉnh u, v , ký hiệu $d_G(u, v)$, là chiều dài nhỏ nhất của một đường đi từ u đến v

Bài toán đường đi ngắn nhất

- **Input:** Đơn đồ thị vô hướng $G = (V, E, w)$ trong đó $V = \{v_0 = a, v_1, \dots, v_n = z\}$, $w : [V]^2 \rightarrow \mathbb{R} \cup \{\infty\}$ với $w(v_i, v_j) = \infty$ nếu $v_i v_j \notin E$
- **Output:** Khoảng cách $d_G(a, z)$

Bài toán đường đi ngắn nhất

Thuật toán Dijkstra



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton
Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

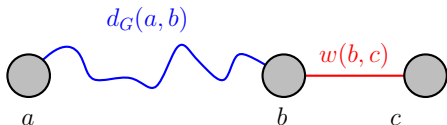
Tô màu đồ thị phẳng

Bài toán đường đi ngắn nhất

- **Input:** Đơn đồ thị vô hướng $G = (V, E, w)$ trong đó $V = \{v_0 = a, v_1, \dots, v_n = z\}$, $w : [V]^2 \rightarrow \mathbb{R} \cup \{\infty\}$ với $w(v_i, v_j) = \infty$ nếu $v_i v_j \notin E$
- **Output:** Khoảng cách $d_G(a, z)$

Thuật toán Dijkstra:

- **Điều kiện:** *Trọng số dương*, nghĩa là $w(v_i, v_j) > 0$ với mọi $0 \leq i < j \leq n$ (hay $w : [V]^2 \rightarrow \mathbb{R}^+ \cup \{\infty\}$)
- **Ý tưởng:** Tìm đường đi ngắn nhất từ a tới các đỉnh kế tiếp cho đến khi đạt đến z . Chú ý rằng với các đỉnh a, b, c , *độ dài đường đi ngắn nhất từ a đến c đi qua đỉnh b kể với c bằng khoảng cách giữa a và b cộng với trọng số cạnh nối b và c*



29

58

Bài toán đường đi ngắn nhất

Thuật toán Dijkstra



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và

Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi ngắn nhất

Đồ thị có trọng số

30

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

“Làm thế nào để tìm được một con đường ngắn nhất để đi từ Rotterdam đến Groningen, hay nói chung: từ một thành phố đến một thành phố khác? Bằng cách sử dụng thuật toán tìm đường đi ngắn nhất, một thuật toán mà tôi đã thiết kế trong khoảng hai mươi phút. Một buổi sáng, tôi đang mua sắm ở Amsterdam cùng với vợ sắp cưới, và khi mệt mỏi, chúng tôi ngồi xuống một quán cà phê để uống một tách cà phê, và tôi chỉ đang nghĩ về việc liệu tôi có thể trả lời được câu hỏi này không, và sau đó tôi đã thiết kế thuật toán tìm đường đi ngắn nhất. Như tôi đã nói, đó là một phát minh chỉ trong có hai mươi phút. Nó được xuất bản vào năm 1959, ba năm sau đó. Bài báo vẫn còn đọc được, và trên thực tế, nó khá hay. Một trong những lý do khiến nó hay đến vậy là vì tôi đã thiết kế nó mà không cần bút và giấy. Sau này tôi mới biết rằng một trong những lợi ích của việc thiết kế mà không cần bút và giấy là bạn gần như buộc phải tránh tất cả những sự phức tạp không cần thiết.”

— Edsger W. Dijkstra, trong một cuộc phỏng vấn với Philip L. Frana, *Communications of the ACM*, 2001

58

Bài toán đường đi ngắn nhất

Thuật toán Dijkstra



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

31 Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

Thuật toán 1: Thuật toán Dijkstra [Dijkstra 1959]

Input: Đơn đồ thị vô hướng $G = (V, E, w)$ trong đó

$$V = \{v_0 = a, v_1, \dots, v_n = z\}, w : [V]^2 \rightarrow \mathbb{R}^+ \cup \{\infty\}$$

với $w(v_i, v_j) = \infty$ nếu $v_i v_j \notin E$

Output: Khoảng cách $d_G(a, z)$

```
1  Gán nhãn  $L(a) := 0, L(v_i) := \infty$  với mọi  $v_i \neq a$ 
2   $S := \emptyset$ 
3  while  $z \notin S$  do
4       $u :=$  đỉnh không thuộc  $S$  với  $L(u)$  nhỏ nhất
5       $S := S \cup \{u\}$ 
6      for  $v \notin S$  do
7          if  $L(u) + w(u, v) < L(v)$  then
8               $L(v) := L(u) + w(u, v)$ 
9  return  $L(z)$ 
```

Bài toán đường đi ngắn nhất

Thuật toán Dijkstra



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

32 Thuật toán Dijkstra

Đồ thị phẳng

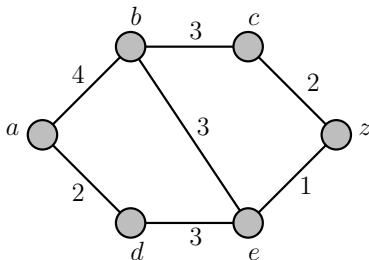
Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

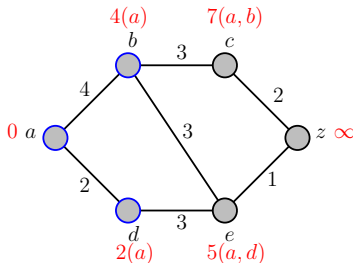
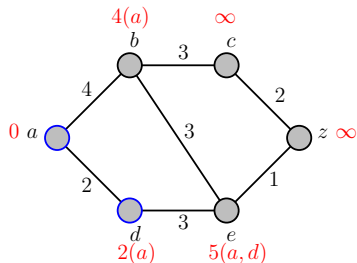
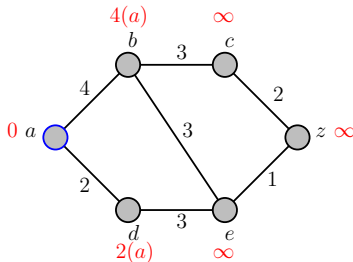
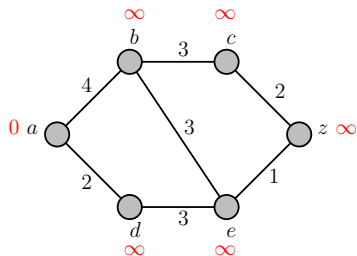
Ví dụ 5

Sử dụng thuật toán Dijkstra để tìm khoảng cách giữa hai đỉnh a và z trong đồ thị sau



Bài toán đường đi ngắn nhất

Thuật toán Dijkstra



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và

Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi ngắn nhất

Đồ thị có trọng số

33

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

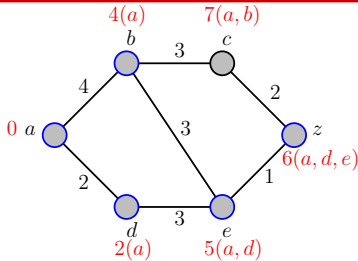
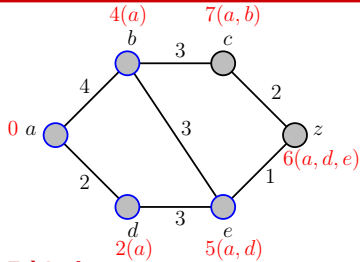
Một số tính chất cơ bản

Tô màu đồ thị phẳng

58

Bài toán đường đi ngắn nhất

Thuật toán Dijkstra



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và

Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi ngắn nhất

Đồ thị có trọng số

34

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

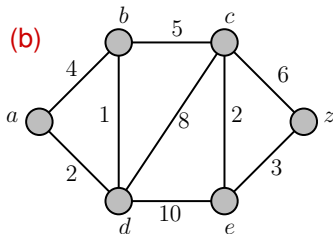
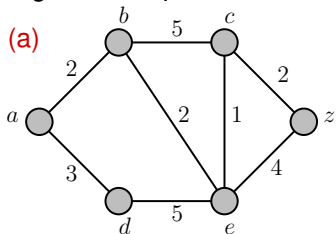
Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

Bài tập 17

Áp dụng thuật toán Dijkstra để tìm khoảng cách từ a đến z trong mỗi đồ thị sau



58

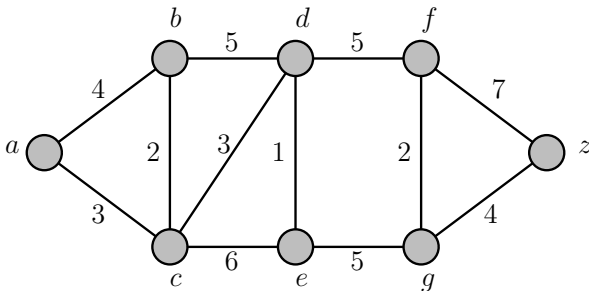
Bài toán đường đi ngắn nhất

Thuật toán Dijkstra



Bài tập 18

Tìm độ dài đường đi ngắn nhất từ a đến z trong đồ thị sau bằng thuật toán Dijkstra



Bài tập 19

Đường đi ngắn nhất giữa hai đỉnh của một đồ thị có trọng số $G = (V, E, w)$ có phải là duy nhất hay không nếu như trọng số của các cạnh là phân biệt, nghĩa là với hai cạnh $e, f \in E$ bất kỳ thì $w(e) \neq w(f)$?

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và

Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi ngắn nhất

Đồ thị có trọng số

35

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

58

Bài toán đường đi ngắn nhất

Thuật toán Dijkstra



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

36

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

- **Tính đúng đắn:** [Dijkstra 1959]
- **Độ phức tạp:** $O(n^2)$, với $n = |V|$
 - Với cấu trúc dữ liệu “đống Fibonacci” (Fibonacci Heap), thuật toán Dijkstra có thể được lập trình để chạy trong thời gian $O(m + n \log n)$, với $n = |V|$ và $m = |E|$
 - Hiệu quả tốt khi chạy với các “đồ thị thưa” (sparse graph) cực lớn (các đồ thị có m rất nhỏ so với n^2)
- Thuật toán tốt nhất được đề xuất gần đây chạy trong thời gian $O(m \log^{2/3} n)$ [Duan et al. 2025]
- Thuật toán Dijkstra cũng có thể được lập trình để xuất ra một đường đi ngắn nhất từ a đến mỗi đỉnh khác trong đồ thị
- Một cách tiếp cận tìm đường đi khác là thuật toán A^* [Hart, Nilsson, and Raphael 1968], thuật toán này mở rộng và cải tiến thuật toán Dijkstra. Thuật toán A^* sử dụng heuristic để định hướng việc tìm kiếm, từ đó đạt hiệu năng tốt hơn

Đồ thị phẳng

Định nghĩa và khái niệm



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton
Đường đi Euler
Đường đi Hamilton

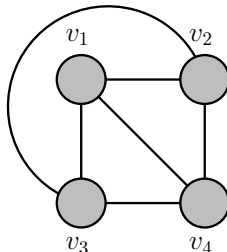
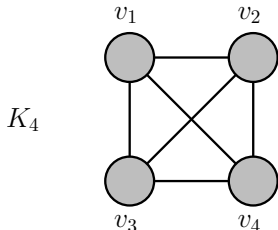
Bài toán đường đi
ngắn nhất
Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng
Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị
Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

- Một đồ thị vô hướng được gọi là **đồ thị phẳng (planar graph)** nếu nó có thể được vẽ trên mặt phẳng sao cho không có hai cạnh nào cắt nhau (ở một điểm không phải là đầu mút của cạnh).
- Hình vẽ như thế được gọi là một **biểu diễn phẳng (planar representation)** của đồ thị.

Ví dụ 6



37

58

Đồ thị phẳng

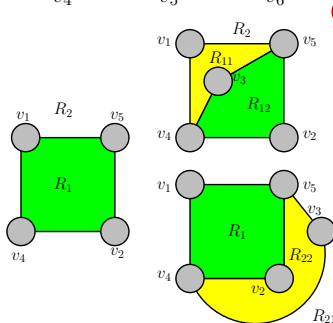
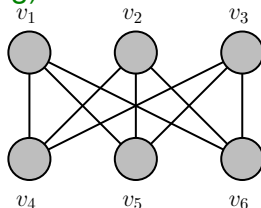
Định nghĩa và khái niệm



Ví dụ 7 ($K_{3,3}$ không là đồ thị phẳng)

Ta chứng minh bằng phản chứng.
Giả sử $K_{3,3}$ là đồ thị phẳng.

- Trong bất kỳ biểu diễn phẳng nào của $K_{3,3}$ ta có v_1 và v_2 đều phải luôn nối với v_4 và v_5 . Các đỉnh này chia mặt phẳng thành hai miền R_1 và R_2 .
- Đỉnh v_3 thuộc R_1 hoặc R_2 .
- Vị trí của v_6 ?



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton
Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất
Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng

38

Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

58

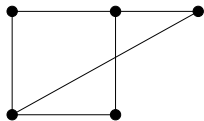
Đồ thị phẳng

Định nghĩa và khái niệm

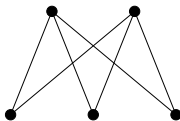


Bài tập 20

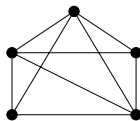
Tìm một biểu diễn phẳng của các đồ thị phẳng sau



G_1



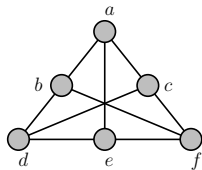
G_2



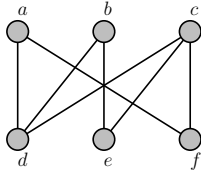
G_3

Bài tập 21

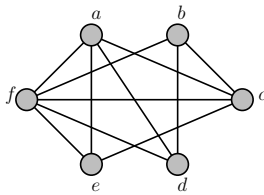
Các đồ thị sau có phải đồ thị phẳng không? Nếu đúng, hãy vẽ một biểu diễn phẳng của đồ thị đó



G_1



G_2



G_3

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng

39

Định nghĩa và khái niệm

Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

58

Đồ thị phẳng

Công thức Euler



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton
Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

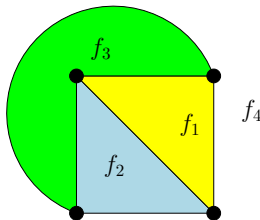
Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

- Biểu diễn phẳng của một đồ thị phẳng $G = (V, E)$ chia mặt phẳng thành các **miền (region)**, kể cả **miền vô hạn (unbounded region)**
- Hai điểm bất kỳ trong cùng một miền có thể được nối với nhau bằng một nét liền mà không cắt bất kỳ cạnh nào
- **Bậc (degree)** của một miền f , ký hiệu $\deg(f)$, là số cạnh của G trên biên của f

Ví dụ 8

Biểu diễn phẳng của K_4

- chia mặt phẳng thành 4 miền f_1, f_2, f_3 , và f_4 ; và
- $\deg(f_1) = \deg(f_2) = \deg(f_3) = \deg(f_4) = 3$



- **Chú ý:** $\sum_{\text{miền } f \text{ của } G} \deg(f) \leq 2|E|$ (Mỗi cạnh thuộc tối đa hai miền)

40

58

Định lý 7: Công thức Euler

Giả sử G là một đơn đồ thị phẳng và liên thông gồm m cạnh, n đỉnh, và r miền. Ta có $n - m + r = 2$

Chứng minh.

- Xây dựng dãy đồ thị $G_1, G_2, \dots, G_m = G$
 - Chọn một cạnh bất kỳ của G làm G_1
 - G_i được tạo thành từ G_{i-1} bằng cách thêm một cạnh bất kỳ liên thuộc với một đỉnh của G_{i-1} ($i \in \{2, 3, \dots, m\}$)
 - Gọi n_i, m_i, r_i lần lượt là số đỉnh, cạnh, và miền của một biểu diễn phẳng của G_i
- Công thức Euler đúng với mọi G_i
 - Ta có $n_1 - m_1 + r_1 = 2 - 1 + 1 = 2$
 - Giả sử công thức Euler đúng với G_i , tức là $n_i - m_i + r_i = 2$
Gọi $a_{i+1}b_{i+1}$ là cạnh thêm vào G_i để tạo thành G_{i+1} . Có hai khả năng:
 - một trong hai đỉnh a_{i+1}, b_{i+1} không thuộc G_{i-1}
 - cả a_{i+1} và b_{i+1} thuộc G_{i-1}

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và

Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

41

58



Hệ quả 8

Giả sử G là một đồ thị phẳng liên thông gồm m cạnh và n đỉnh ($n \geq 3$). Khi đó, $m \leq 3n - 6$. Thêm vào đó, nếu $m = 3n - 6$ thì mỗi miền của G có chính xác 3 cạnh trên biên.

Chứng minh.

- Nhận xét rằng mỗi miền của G có ít nhất 3 cạnh trên biên,

do đó $\sum_{\text{miền } f \text{ của } G} \deg(f) \geq 3r$. Mặt khác, ta cũng có

$$\sum_{\text{miền } f \text{ của } G} \deg(f) \leq 2m. \text{ Suy ra } 3r \leq 2m.$$

- Áp dụng công thức Euler, ta có

$$2 = n - m + r \leq n - m + 2m/3, \text{ suy ra } m \leq 3n - 6.$$



Bài tập 22

Giả sử G là một đồ thị đơn phẳng và liên thông gồm 20 đỉnh, mỗi đỉnh có bậc 3. Một biểu diễn phẳng của G chia mặt phẳng thành bao nhiêu miền?

Bài tập 23

Trong số các đồ thị không phẳng sau, đồ thị nào thỏa mãn điều kiện sau: nếu bỏ đi một đỉnh bất kỳ và các cạnh liên thuộc với đỉnh đó thì ta thu được một đồ thị phẳng.

- (a) K_5 (c) $K_{3,3}$
(b) K_6 (d) $K_{3,4}$

Bài tập 24

Chứng minh K_5 không là đồ thị phẳng

Bài tập 25

Chứng minh rằng nếu G là một đơn đồ thị phẳng và liên thông thì G có một đỉnh có bậc nhỏ hơn hoặc bằng 5

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

43

58

Đồ thị phẳng

Định lý Kuratowski



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

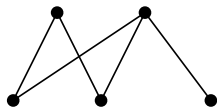
Giới thiệu

Một số tính chất cơ bản

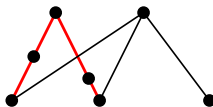
Tô màu đồ thị phẳng

- Cho đồ thị G . Một **phép phân chia (subdivision)** một cạnh e của G được thực hiện bằng cách thay thế e bằng một đường đi đơn
- Hai đồ thị G_1 và G_2 được gọi là **đồng phôi (homeomorphic)** nếu chúng được xây dựng từ cùng một đồ thị thông qua một dãy các phép phân chia

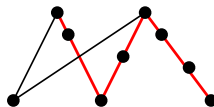
Ví dụ 9



G



G_1



G_2

Định lý 9: Định lý Kuratowski

G là đồ thị phẳng khi và chỉ khi nó không chứa bất kỳ đồ thị nào đồng phôi với K_5 hoặc $K_{3,3}$

44

58



Có thể hình dung định lý Kuratowski một cách sơ lược như sau:

- Chúng ta biết hai đồ thị không phẳng, đó là $K_{3,3}$ và K_5 . Do đó, đương nhiên bất kỳ đồ thị nào chứa những đồ thị này đều không phải là đồ thị phẳng
- Thực tế, bất kỳ đồ thị nào chứa cấu trúc có hình dạng cơ bản giống như hai đồ thị trên đều không phẳng (liên quan đến các khái niệm phép phân chia (subdivision) và đồng phôi (homeomorphic))
- Hơn nữa, mọi đồ thị không phẳng đều chứa một trong hai “hình dạng xấu” này. Đây là cách duy nhất để một đồ thị trở thành không phẳng

Đường đi Euler và
Đường đi Hamilton
Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất
Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng
Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị
Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

Đồ thị phẳng

Định lý Kuratowski



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm
Công thức Euler

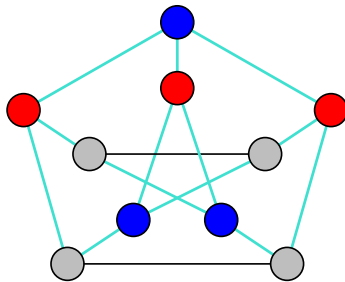
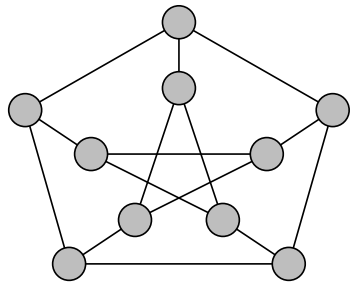
Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng



Hình: Đồ thị Petersen có chứa một đồ thị con đồng phôi với $K_{3,3}$ và do đó không phải là đồ thị phẳng

46

58

Tô màu đồ thị

Giới thiệu



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

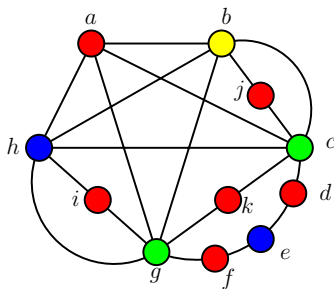
47

Giới thiệu

Một số tính chất cơ bản
Tô màu đồ thị phẳng

Cho đơn đồ thị vô hướng $G = (V, E)$

- **Tô màu** một đồ thị đơn là sự gán màu cho các đỉnh của đồ thị sao cho không có hai đỉnh liên kề được gán cùng một màu. Cụ thể, với các “màu” $1, 2, \dots, k$, một **cách tô màu các đỉnh (vertex k -coloring)** của G là một hàm $f : V \rightarrow \{1, 2, \dots, k\}$ thỏa mãn $f(u) \neq f(v)$ với mọi $u, v \in V$ với $uv \in E$



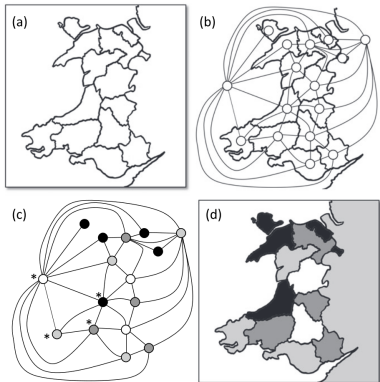
- **Sắc số (chromatic number)** của G , ký hiệu $\chi(G)$, là số tối thiểu các màu cần thiết để tô màu G

58

Tô màu đồ thị

Giới thiệu

Ghi chép sớm nhất về bài toán tô màu đồ thị có lẽ là vào năm 1852 khi Francis Guthrie (1831–1899), lúc đó là một sinh viên ở Đại học Cao đẳng London (University College London), tô màu một bản đồ các quận của Anh và nhận ra là có lẽ chỉ cần bốn màu để tô màu bản đồ sao cho hai quận liền kề nhau có màu khác nhau



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

48

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

58

Tô màu đồ thị

Giới thiệu



- Phỏng đoán của Guthrie được cho là phát biểu đầu tiên của **Định lý bốn màu (Four Color Theorem)**

Định lý 10: Định lý bốn màu

Với mọi đồ thị phẳng G , ta luôn có $\chi(G) \leq 4$

- Năm 1879, Kempe đề xuất một chứng minh cho Định lý bốn màu. Khoảng 10 năm sau đó, Heawood chỉ ra lỗi sai trong chứng minh của Kempe và chỉnh sửa lại chứng minh của Kempe để chỉ ra rằng năm màu là đủ để tô màu bất kỳ đồ thị phẳng nào
- Năm 1976, Kenneth Appel and Wolfgang Haken (Đại học Illinois) [Appel and Haken 1977]; [Appel, Haken, and Koch 1977] chứng minh định lý bốn màu bằng cách giả sử nếu Định lý bốn màu là sai thì sẽ có một phản ví dụ thuộc một trong 1936 loại khác nhau, và chỉ ra rằng không có loại nào dẫn đến phản ví dụ. Các trường hợp này được phân tích cẩn thận nhờ máy tính
- Robertson, Sanders, Seymour, và Thomas [Robertson, Sanders, Seymour, and Thomas 1997] đưa ra một chứng minh đơn giản hơn với 633 loại cần kiểm tra

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

49

Giới thiệu

Một số tính chất cơ bản
Tô màu đồ thị phẳng

Tô màu đồ thị

Một số tính chất cơ bản



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

- Để **chỉ ra** $\chi(G) = k$ với đồ thị G nào đó, ta cần:
 - Chỉ ra một cách tô màu các đỉnh của G bằng k màu (nghĩa là $\chi(G) \leq k$)
 - Chỉ ra rằng không thể dùng ít hơn k màu để tô màu các đỉnh của G (nghĩa là $\chi(G) \geq k$)
- Một số nhận xét
 - (1) Mọi đồ thị G gồm n đỉnh có thể được tô màu bằng n màu
 - (2) $\chi(K_n) = n$ (Tại sao?)
 - (3) Ta ký hiệu $\omega(G)$ là số nguyên dương lớn nhất $r \geq 1$ thỏa mãn K_r là đồ thị con của G . Với mọi đồ thị G , ta có $\omega(G) \leq \chi(G)$. Thông thường, $\omega(G) \neq \chi(G)$
 - (4) $\chi(C_n) = 2$ nếu $n \geq 4$ chẵn và $\chi(C_n) = 3$ nếu $n \geq 3$ lẻ (Tại sao?)
 - (5) G là đồ thị hai phần khi và chỉ khi $\chi(G) \leq 2$ (Tại sao?)
- Chưa biết có tồn tại hay không một thuật toán chạy trong thời gian đa thức để xác định xem một đồ thị G có thể được tô màu bằng 3 màu hay không

Bài tập 26

Tính $\chi(W_n)$, $\chi(K_{m,n})$, và $\chi(Q_n)$

50

58

Tô màu đồ thị

Một số tính chất cơ bản



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

Bài tập 27

Những đồ thị nào có sắc số bằng 1?

Bài tập 28

Một cách tô màu các cạnh của một đồ thị $G = (V, E)$ bằng k màu (*k -edge coloring*) là một hàm $f : E \rightarrow \{1, 2, \dots, k\}$ thỏa mãn điều kiện $f(e) \neq f(e')$ nếu e và e' liên thuộc với cùng một đỉnh. *Sắc số cạnh (edge chromatic number)* của một đồ thị G , ký hiệu $\chi'(G)$, là số màu nhỏ nhất có thể dùng để tô màu các cạnh của G

- (a) Tìm $\chi'(C_n)$ và $\chi'(W_n)$ với $n \geq 3$
- (b) Chứng minh rằng $\chi'(G) \geq \Delta(G)$, trong đó $\Delta(G)$ là bậc lớn nhất của một đỉnh của G

51

58

Tô màu đồ thị

Một số tính chất cơ bản



Gọi $\Delta(G)$ là *bậc lớn nhất của các đỉnh của đồ thị G*

Định lý 11

Cho $G = (V, E)$ là đơn đồ thị vô hướng có n đỉnh. Ta có $\chi(G) \leq \Delta(G) + 1$

Chứng minh.

Một thuật toán tham lam để tô màu các đỉnh của G bằng $\Delta(G) + 1$ màu $\{1, \dots, \Delta(G) + 1\}$ là như sau:

1. Gán nhãn $v_1, v_2, \dots, v_n$ cho các đỉnh của G một cách tùy ý
2. Với i từ 1 đến n , tô màu đỉnh v_i bằng màu nhỏ nhất trong số các màu chưa được tô cho bất kỳ đỉnh nào trong $N(v_i)$

Ta chứng minh rằng *Bước 2 của thuật toán luôn thực hiện được với $\Delta(G) + 1$ màu*. Thật vậy, đỉnh v_i có tối đa $\Delta(G)$ đỉnh kề với nó, do đó số màu tối đa sử dụng để tô màu các đỉnh trong $N(v_i)$ là $\Delta(G)$, nghĩa là luôn có ít nhất một trong số $\Delta(G) + 1$ màu không được sử dụng cho bất kỳ đỉnh nào kề với v_i , và ta có thể tô màu v_i bằng màu nhỏ nhất trong số các màu này $\square$

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và

Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

52

Một số tính chất cơ bản

Tô màu đồ thị phẳng

58

Tô màu đồ thị

Tô màu đồ thị phẳng



Bổ đề 13

Mọi đơn đồ thị phẳng và liên thông G gồm n đỉnh có một cách sắp thứ tự các đỉnh $v_1, v_2, \dots, v_n$ sao cho mỗi đỉnh kề với tối đa 5 đỉnh đứng trước nó

Chứng minh.

Ta chứng minh bằng quy nạp theo n .

- **Bước cơ sở:** Với $n \leq 6$, bất kể thứ tự sắp xếp các đỉnh nào đều thỏa mãn Bổ đề
- **Bước quy nạp:**
 - Giả sử Bổ đề đúng với mọi $6 \leq n \leq k$, trong đó $k \geq 6$ là một số nguyên nào đó. Ta chứng minh Bổ đề đúng với $n = k + 1$.
 - Thật vậy, giả sử G là đồ thị bất kỳ gồm $k + 1$ đỉnh. Từ Bài tập 25, tồn tại một đỉnh v của G thỏa mãn $\deg(v) \leq 5$.
 - Đồ thị $G - v$:
 - có tối đa 5 thành phần liên thông $G_1, G_2, \dots, G_5$
 - mỗi G_i có $n_i \leq k$ đỉnh ($1 \leq i \leq 5$)
 - và $n_1 + \dots + n_5 = k$
 - Từ giả thiết quy nạp, tồn tại một thứ tự $v_1, \dots, v_k$ các đỉnh của $G - v$ thỏa mãn Bổ đề.
 - Đặt $v_{k+1} = v$, ta có $v_1, \dots, v_k, v_{k+1}$ là một thứ tự các đỉnh của G thỏa mãn Bổ đề

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

54

58

Tô màu đồ thị

Tô màu đồ thị phẳng



Định lý 14

Mọi đồ thị phẳng G có $\chi(G) \leq 6$

Chứng minh.

Ta chỉ ra một cách tô màu đơn đồ thị phẳng và liên thông G bằng 6 màu

1. Tìm thứ tự các đỉnh $v_1, v_2, \dots, v_n$ của G thỏa mãn Bổ đề 13: mỗi đỉnh có tối đa 5 đỉnh kề đứng trước nó
2. Áp dụng thuật toán tham lam ở Định lý 11 với thứ tự đỉnh tìm được ở Bước 1

Chú ý rằng

- Khuyên và cạnh song song (nếu có) không ảnh hưởng gì đến quá trình tô màu
- Nếu đồ thị phẳng đã cho không liên thông, ta có thể áp dụng quá trình tô màu riêng biệt cho từng thành phần liên thông

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton
Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất
Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng
Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị
Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

55

58

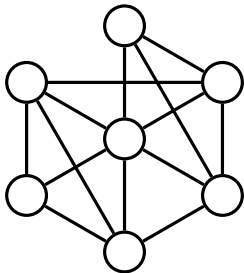
Tô màu đồ thị

Tô màu đồ thị phẳng

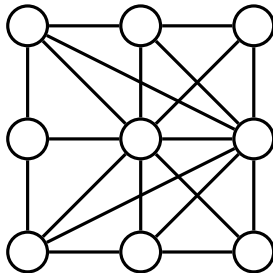


Bài tập 30

Sử dụng thuật toán ở Định lý 14 để tô màu các đồ thị phẳng sau bằng 6 màu $\{1, 2, 3, 4, 5, 6\}$



Đồ thị G



Đồ thị H

Nếu có thể, hãy tìm một cách tô màu các đồ thị trên bằng 5 màu hoặc ít hơn

Bài tập 31

Chứng minh Định lý 14 bằng phương pháp quy nạp

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và

Đường đi Hamilton

Đường đi Euler

Đường đi Hamilton

Bài toán đường đi ngắn nhất

Đồ thị có trọng số

Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm

Công thức Euler

Định lý Kuratowski

Tô màu đồ thị

Giới thiệu

Một số tính chất cơ bản

Tô màu đồ thị phẳng

56

58

Tô màu đồ thị

Tô màu đồ thị phẳng



Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và
Đường đi Hamilton

Đường đi Euler
Đường đi Hamilton

Bài toán đường đi
ngắn nhất

Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng

Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị

Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

Định lý 15

Mọi đồ thị phẳng G có $\chi(G) \leq 5$

Bài tập 32

Bài tập sau đề xuất một cách chứng minh Định lý 15 bằng phương pháp quy nạp theo số đỉnh n của G

- (a) Ở bước cơ sở, chứng minh Định lý 15 đúng với mọi $n \leq 5$
- (b) Ở bước quy nạp, giả sử Định lý 15 đúng với mọi đồ thị phẳng gồm n đỉnh, trong đó $n \geq 5$ là số nguyên nào đó. Chứng minh Định lý 15 cũng đúng với mọi đồ thị phẳng gồm $n + 1$ đỉnh. Giả sử G là đồ thị phẳng với $n + 1$ đỉnh. Từ Bài tập 25, tồn tại một đỉnh v của G thỏa mãn $\deg(v) \leq 5$. Theo giả thiết quy nạp, có một cách tô màu α các đỉnh của $G - v$ bằng 5 màu $\{1, 2, 3, 4, 5\}$



Bài tập 32 (tiếp)

(b.1) Nếu $\deg(v) \leq 4$ hoặc $\deg(v) = 5$ và các đỉnh kề với v được tô màu bằng ≤ 4 màu, chứng minh rằng các đỉnh của G có thể được tô màu bằng 5 màu

(b.2) (★) Nếu $\deg(v) = 5$ và các đỉnh kề với v , ví dụ v_1, v_2, v_3, v_4, v_5 , được tô màu bằng 5 màu khác nhau, ví dụ như lần lượt là 1, 2, 3, 4, 5, chứng minh rằng các đỉnh của G có thể được tô màu bằng 5 màu

- Xét đồ thị $G_{1,3}$ cảm sinh bởi các đỉnh tô màu 1 và 3. Điều gì xảy ra nếu v_1 và v_3 không thuộc cùng một thành phần liên thông của $G_{1,3}$? (**Gợi ý:** Trong trường hợp này, liệu có cách tô màu nào để v_1 và v_3 có cùng màu không?)
- Nếu v_1 và v_3 thuộc cùng một thành phần liên thông của $G_{1,3}$, xét đồ thị $G_{2,4}$ cảm sinh bởi các đỉnh tô màu 2 và 4. Liệu có thể xảy ra trường hợp v_2 và v_4 thuộc cùng một thành phần liên thông của $G_{2,4}$ không? (**Gợi ý:** G là đồ thị phẳng)

Part I

Phụ lục

Nội dung



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

Bài toán đường đi ngắn nhất
Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng
Đường đi Euler
Đường đi Hamilton

Bài toán đường đi ngắn nhất

Thuật toán Floyd



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

Thuật toán Floyd:

■ Lịch sử:

- Được phát triển bởi Robert W. Floyd vào năm 1962 [Floyd 1962]
- Tuy nhiên, nó về cơ bản giống với các thuật toán đã được công bố trước đó bởi Bernard Roy vào năm 1959 [Roy 1959] và cũng bởi Stephen Warshall vào năm 1962 [Warshall 1962] để tìm bao đóng bắc cầu (transitive closure) của đồ thị

■ Điều kiện:

- Trọng số có thể âm
- *Không có chu trình có tổng trọng số âm*

■ Ý tưởng:

- Xét đường đi từ v_j đến v_k chỉ qua các đỉnh $\{v_0, v_1, \dots, v_i\}$
- Nếu đường đi từ v_j đến v_k qua v_i ngắn hơn đường đi hiện tại, ta cập nhật khoảng cách

2

Bài toán đường đi ngắn nhất

Thuật toán Floyd



Thuật toán 2: Thuật toán Floyd [Floyd 1962]

Input: Đơn đồ thị vô hướng $G = (V, E, w)$ trong đó

$V = \{v_0 = a, v_1, \dots, v_n = z\}$, $w : [V]^2 \rightarrow \mathbb{R} \cup \{\infty\}$ với
 $w(v_i, v_j) = \infty$ nếu $v_i v_j \notin E$, và G không có chu trình
có tổng trọng số âm

Output: Khoảng cách $d_G(v_i, v_j)$ với mọi $0 \leq i < j \leq n$

```
1 for i := 0 to n do
2   for j := 0 to n do
3     d(v_i, v_j) := w(v_i, v_j)
4 for i := 0 to n do
5   for j := 0 to n do
6     for k := 0 to n do
7       if d(v_j, v_i) + d(v_i, v_k) < d(v_j, v_k) then
8         d(v_j, v_k) := d(v_j, v_i) + d(v_i, v_k)
9 return {d(v_i, v_j) | i, j ∈ {0, 1, ..., n}}
```

Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi
ngắn nhất

3

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

Bài toán đường đi ngắn nhất

Thuật toán Floyd: Ví dụ



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

4 Thuật toán Floyd

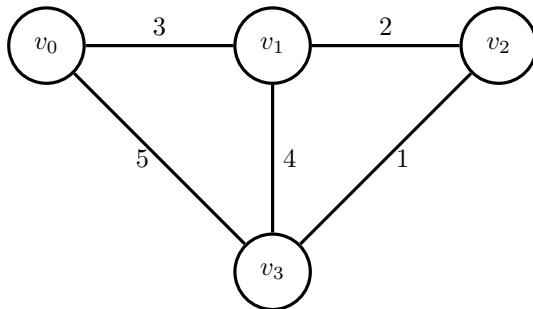
Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo



Hình: Đồ thị ví dụ cho thuật toán Floyd

Bài toán đường đi ngắn nhất

Thuật toán Floyd: Khởi tạo



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

- Khởi tạo ma trận khoảng cách $d(v_i, v_j) = w(v_i, v_j)$:

d	v_0	v_1	v_2	v_3
v_0	0	3	∞	5
v_1	3	0	2	4
v_2	∞	2	0	1
v_3	5	4	1	0

- Các giá trị ∞ cho biết không có cạnh trực tiếp nối hai đỉnh
- Thuật toán sẽ xét lần lượt các đỉnh trung gian i từ v_0 đến v_3

Bài toán đường đi ngắn nhất

Thuật toán Floyd: Lặp với $i = 0$ (v_0)



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

6 Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

- Xét v_0 làm đỉnh trung gian, kiểm tra $d(v_j, v_0) + d(v_0, v_k) < d(v_j, v_k)$?
- Không có cải thiện khoảng cách khi xét v_0 làm đỉnh trung gian

d	v_0	v_1	v_2	v_3
v_0	0	3	∞	5
v_1	3	0	2	4
v_2	∞	2	0	1
v_3	5	4	1	0

Bài toán đường đi ngắn nhất

Thuật toán Floyd: Lặp với $i = 1$ (v_1)



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

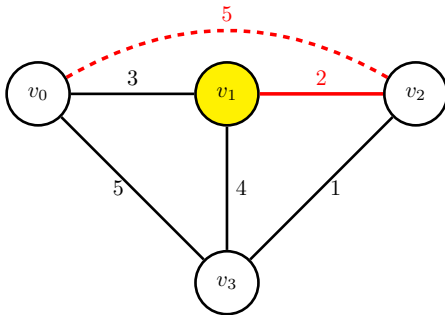
Tài liệu tham khảo

- Xét v_1 làm đỉnh trung gian, kiểm tra $d(v_j, v_1) + d(v_1, v_k) < d(v_j, v_k)$?

- Tìm được cải thiện:

$$d(v_0, v_1) + d(v_1, v_2) = 3 + 2 = 5 < \infty = d(v_0, v_2)$$

d	v_0	v_1	v_2	v_3
v_0	0	3	5	5
v_1	3	0	2	4
v_2	5	2	0	1
v_3	5	4	1	0



Bài toán đường đi ngắn nhất

Thuật toán Floyd: Lặp với $i = 2$ (v_2)



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

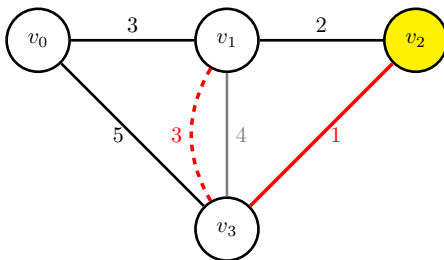
- Xét v_2 làm đỉnh trung gian, kiểm tra

$$d(v_j, v_2) + d(v_2, v_k) < d(v_j, v_k)?$$

- Tìm được cải thiện:

$$d(v_1, v_2) + d(v_2, v_3) = 2 + 1 = 3 < d(v_1, v_3) = 4$$

d	v_0	v_1	v_2	v_3
v_0	0	3	5	5
v_1	3	0	2	3
v_2	5	2	0	1
v_3	5	3	1	0



Bài toán đường đi ngắn nhất

Thuật toán Floyd: Lặp với $i = 3$ (v_3)



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

- Xét v_3 làm đỉnh trung gian, kiểm tra $d(v_j, v_3) + d(v_3, v_k) < d(v_j, v_k)$?
- Không có cải thiện khoảng cách khi xét v_3 làm đỉnh trung gian

d	v_0	v_1	v_2	v_3
v_0	0	3	5	5
v_1	3	0	2	3
v_2	5	2	0	1
v_3	5	3	1	0

- Ma trận khoảng cách cuối cùng cho kết quả khoảng cách giữa mọi cặp đỉnh

Bài toán đường đi ngắn nhất

Thuật toán Floyd: Phân tích



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

10 Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

- **Tính đúng đắn:** [Floyd 1962]

- **Độ phức tạp:** $O(n^3)$ với $n = |V|$

- **Ưu điểm:**

- Tìm khoảng cách giữa mọi cặp đỉnh
- Có thể xử lý đồ thị có cạnh trọng số âm (không có chu trình âm)

- **Nhược điểm:**

- Chậm hơn thuật toán Dijkstra khi chỉ cần tìm đường đi từ một nguồn
- Thuật toán có thể mở rộng để tìm đường đi ngắn nhất (không chỉ khoảng cách)

Bài toán đường đi ngắn nhất

Thuật toán Floyd: Câu hỏi



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

11 Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

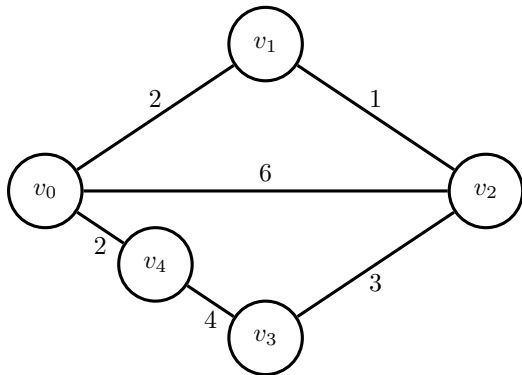
Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

Bài tập 33

Áp dụng thuật toán Floyd để tìm khoảng cách giữa mọi cặp đỉnh trong đồ thị sau:



Một số lỗi thường gặp



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

12 Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

Chú ý

Tham khảo từ tài liệu “Common Mistakes in Discrete Mathematics” (https://highered.mheducation.com/sites/dl/free/125967651x/1106131/Common_Mistakes_in_Discrete_Math.pdf)

(a) Nhầm lẫn giữa định nghĩa đường đi và chu trình Hamilton và Euler

- Hãy nhớ rằng có một phương pháp kiểm tra đơn giản cho đường đi và chu trình Euler, nhưng không có phương pháp kiểm tra đơn giản nào cho đường đi và chu trình Hamilton

(b) Bỏ qua thực tế rằng việc có một chu trình Euler [Hamilton] hàm ý sự tồn tại của một đường đi Euler [Hamilton]

- Hãy xem xét kỹ các định nghĩa

Một số lỗi thường gặp (tiếp)



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

13 Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

(c) Nhầm lẫn giữa các định lý trong lý thuyết đồ thị với mệnh đề đảo của chúng

- Ví dụ, nếu trong một đồ thị đơn liên thông với $n \geq 3$ đỉnh, mỗi đỉnh có bậc ít nhất $n/2$, thì đồ thị đó có một chu trình Hamilton; nhưng mệnh đề đảo hoặc mệnh đề ngược của phát biểu này không đúng (có nhiều đồ thị có chu trình Hamilton mà trong đó bậc của các đỉnh nhỏ).
- Đây là một ví dụ khác: Nếu một đồ thị đơn liên thông là đồ thị phẳng, thì nó phải thỏa mãn $e \leq 3v - 6$, trong đó e là số cạnh và v là số đỉnh. Do đó (theo phản đảo), ta biết rằng nếu một đồ thị có quá nhiều cạnh ($e > 3v - 6$), thì nó không thể là đồ thị phẳng. Điều mà chúng ta không thể kết luận là mệnh đề đảo—không phải là một định lý—rằng nếu $e \leq 3v - 6$, thì đồ thị đó phải là đồ thị phẳng

(d) Nhầm tưởng rằng một đồ thị không phẳng chỉ vì nó được vẽ theo cách mà có hai cạnh cắt nhau tại các điểm không phải là đầu mút

Một số lỗi thường gặp (tiếp)



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

14 Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

- Nếu có thể vẽ lại đồ thị đó mà không có cạnh nào cắt nhau ở các điểm không phải là đầu mút, thì đồ thị đó là đồ thị phẳng
- Ví dụ, K_4 là đồ thị phẳng, mặc dù khi vẽ nó với các đỉnh là đỉnh của một hình vuông và các đoạn thẳng biểu diễn các cạnh gây ra giao điểm ở giữa hình (Hãy vẽ lại nó dưới dạng các đỉnh của một tam giác với một đỉnh nữa ở bên trong)

(e) **Sai lầm khi kết luận rằng một khi đã tìm được cách tô màu đồ thị với n màu, thì số màu cần dùng của đồ thị đó phải là n**

- Thực tế, điều chúng ta biết trong trường hợp đó chỉ là số màu cần dùng nhiều nhất là n
- Có thể tìm được cách tô màu khác với ít hơn n màu
- Ví dụ, người ta có thể tô C_4 bằng bốn màu (một màu khác nhau cho mỗi đỉnh), nhưng số màu cần dùng thực sự là 2

(f) **Nhầm tưởng rằng thuật toán tham lam luôn đưa ra giải pháp tối ưu cho một bài toán**

Một số lỗi thường gặp (tiếp)



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

15 Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

- Mặc dù trong nhiều trường hợp giải thuật tham lam đơn giản có thể tìm ra giải pháp tốt nhất (ví dụ, trong việc tìm kiếm cây khung nhỏ nhất), nhưng thường thì giải thuật tham lam hay thất bại (ví dụ, trong việc tìm cách tô màu đồ thị sử dụng ít màu nhất có thể)

(g) Không nhận ra rằng việc viết ra một thuật toán không đảm bảo rằng thuật toán đó làm những gì bạn muốn

- Ví dụ, không thể viết một thuật toán tham lam để tô màu đồ thị và sau đó khẳng định mà không có sự chứng minh rằng thuật toán này tìm ra cách tô màu với số lượng màu ít nhất có thể

Đường đi Euler

Ứng dụng: Ghép các mảnh DNA



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

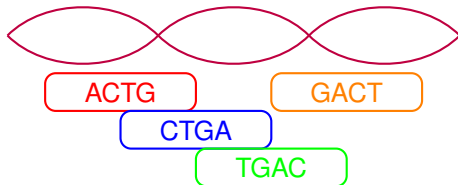
Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

- **Bài toán giải trình tự DNA:** Xác định trình tự nucleotide trong một phân tử DNA
- **Thách thức:** Các thiết bị chỉ đọc được các đoạn ngắn (reads), cần ghép các đoạn này thành chuỗi hoàn chỉnh
- **Phương pháp:** Sử dụng đồ thị de Bruijn và đường đi Euler để tái tạo trình tự ban đầu



Hình: Đọc các đoạn mã di truyền ngắn từ chuỗi DNA

16

33

Đường đi Euler

Đồ thị de Bruijn và k -mers



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi
ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

17

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

■ **k -mer:** Các đoạn con có độ dài k của chuỗi DNA

■ **Đồ thị de Bruijn:**

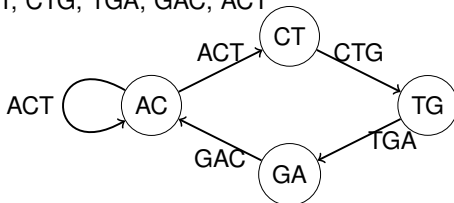
■ **Đỉnh:** Các $(k - 1)$ -mers

■ **Cạnh:** Từ $(k - 1)$ -mer u đến $(k - 1)$ -mer v nếu có k -mer trong đó u là tiền tố và v là hậu tố

■ **Nhãn cạnh:** k -mer tương ứng

Chuỗi ban đầu: ACTGACT

3-mers: ACT, CTG, TGA, GAC, ACT



Hình: Đồ thị de Bruijn cho chuỗi ACTGACT với $k = 3$

Đường đi Euler

Tái tạo chuỗi DNA bằng đường đi Euler



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi
ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

18

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

- **Lý thuyết:** Chuỗi DNA ban đầu tương ứng với một đường đi Euler trong đồ thị de Bruijn
- **Phương pháp:**
 - Tạo đồ thị de Bruijn từ tập các k -mers
 - Tìm đường đi Euler trong đồ thị
 - Tái tạo chuỗi từ đường đi tìm được

DNA gốc (không biết)

Đoạn 1 Đoạn 2 Đoạn 3 Đoạn 4 Đoạn 5 Đoạn 6 Đoạn 7



Đồ thị de Bruijn



DNA tái tạo

Hình: Quy trình tái tạo chuỗi DNA sử dụng đồ thị de Bruijn

Đường đi Euler

Ví dụ minh họa



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi
ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

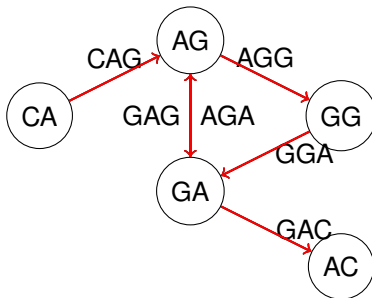
Tài liệu tham khảo

■ Các đoạn đọc:

- CAGGA
- AGGAG
- GGAGA
- GAGAC

■ 3-mers:

- CAG, AGG, GGA
- AGG, GGA, GAG
- GGA, GAG, AGA
- GAG, AGA, GAC



Hình: Đồ thị de Bruijn với $k = 3$

Đường đi Euler: $CA \rightarrow AG \rightarrow GG \rightarrow GA \rightarrow AG \rightarrow GA \rightarrow AC$
Chuỗi tái tạo: CAGGAGAC

Đường đi Euler

Ưu và nhược điểm của phương pháp



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

Ưu điểm:

- Tính toán hiệu quả
- Xử lý được dữ liệu lớn
- Cơ sở toán học chặt chẽ
- Phương pháp thông dụng trong các công cụ lắp ráp trình tự DNA

Thách thức:

- Xử lý lỗi trong dữ liệu đọc
- Các trình tự lặp lại
- Sự tồn tại của nhiều đường đi Euler khác nhau
- Cần thêm thông tin bổ sung để xác định đường đi chính xác

20

33

Đường đi Euler

Ứng dụng: Bài toán người đưa thư Trung Hoa



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi
ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

21

■ Bài toán người đưa thư Trung Hoa (Chinese Postman Problem - CPP):

- Đề xuất bởi nhà toán học Trung Quốc Mei-Ko Kwan (Quản Mai Cốc) năm 1962. Tên gọi của bài toán được đề xuất bởi Alan Goldman (Cục Tiêu chuẩn quốc gia Hoa Kỳ)
- Mô hình hóa việc một người đưa thư cần đi qua tất cả các con đường trong khu vực
- Người đưa thư muốn quay lại điểm xuất phát với quãng đường đi ngắn nhất có thể

■ Mối liên hệ với đường đi Euler:

- Nếu đồ thị có chu trình Euler: người đưa thư chỉ cần đi mỗi con đường đúng một lần
- Nếu đồ thị không có chu trình Euler: người đưa thư phải đi qua một số con đường nhiều hơn một lần

- **Ứng dụng:** Tối ưu hóa lộ trình giao hàng, thu gom rác thải, bảo trì đường phố, v.v.

Đường đi Euler

Ứng dụng: Bài toán người đưa thư Trung Hoa



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

- **Đầu vào:** Đồ thị $G = (V, E)$ có trọng số (có thể có hướng hoặc vô hướng)
- **Đầu ra:** Chu trình với tổng trọng số nhỏ nhất đi qua mỗi cạnh ít nhất một lần
- **Giải thuật cho đồ thị vô hướng:**
 1. Kiểm tra tính liên thông của đồ thị. Nếu không liên thông, bài toán không có lời giải.
 2. Xác định tập O các đỉnh bậc lẻ trong đồ thị. ($|O|$ phải là số chẵn)
 3. Nếu $|O| = 0$ (không có đỉnh bậc lẻ):
 - Đồ thị đã có chu trình Euler, đây chính là lời giải tối ưu.
 4. Nếu $|O| > 0$ (có đỉnh bậc lẻ):
 - Với mỗi cặp đỉnh $u, v \in O$, tính khoảng cách $d(u, v)$ giữa chúng
 - Tìm bộ ghép hoàn hảo trọng số nhỏ nhất giữa các đỉnh trong O
 - Thêm các cạnh tương ứng với các đường đi ngắn nhất từ bộ ghép vào đồ thị ban đầu
 5. Tìm chu trình Euler trong đồ thị đã được bổ sung các cạnh và trả lại chu trình đó

22

33

Đường đi Euler

Ví dụ minh họa



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

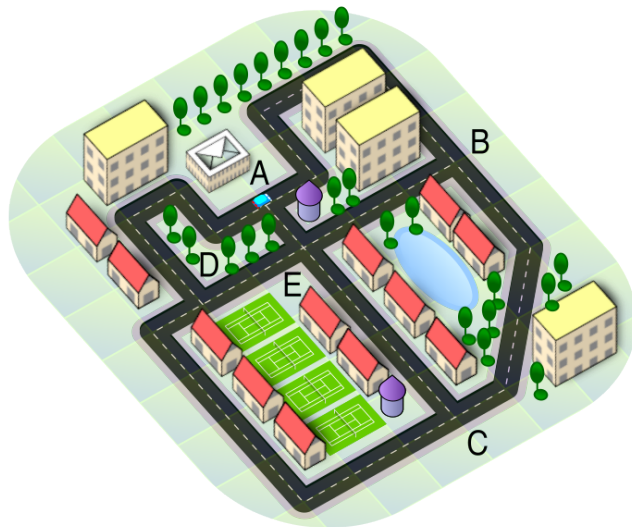
Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

23



Hình: Minh họa Bài toán người đưa thư Trung Hoa (ví dụ từ Wikipedia)

33

Đường đi Euler

Ví dụ minh họa



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

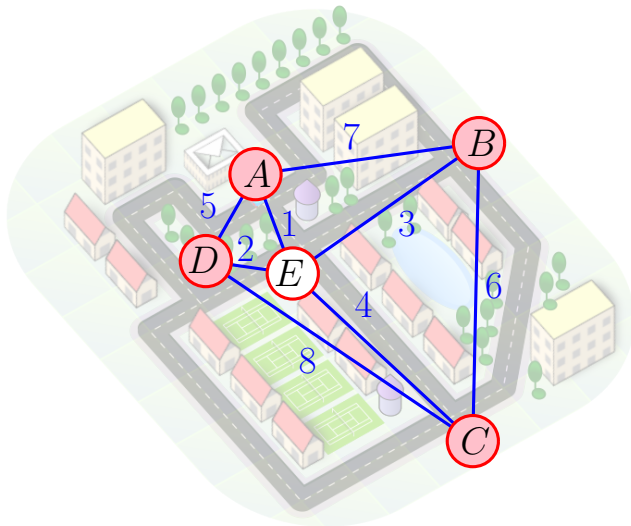
Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

23



Hình: Minh họa Bài toán người đưa thư Trung Hoa (ví dụ từ Wikipedia)

33

Đường đi Euler

Ví dụ minh họa



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi
ngắn nhất

Thuật toán Floyd

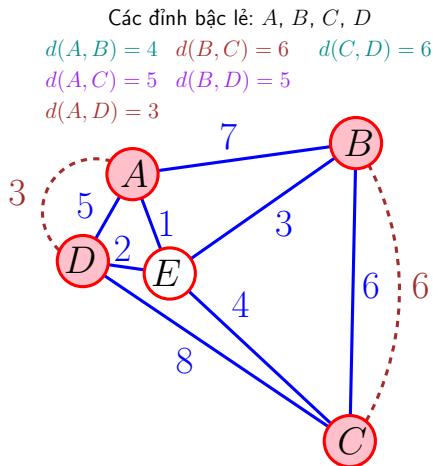
Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo



Hình: Minh họa Bài toán người đưa thư Trung Hoa (ví dụ từ Wikipedia)

Đường đi Euler

Ví dụ minh họa



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi
ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

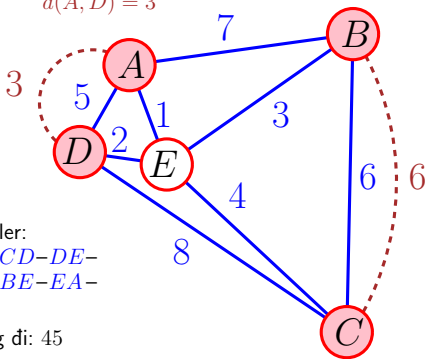
Tài liệu tham khảo

Các đỉnh bậc lẻ: A, B, C, D

$$d(A, B) = 4 \quad d(B, C) = 6 \quad d(C, D) = 6$$

$$d(A, C) = 5 \quad d(B, D) = 5$$

$$d(A, D) = 3$$



Chu trình Euler:

$AB-BC-CD-DE-$
 $EC-CB-BE-EA-$
 $AD-DA$

Độ dài đường đi: 45

23

33

Đường đi Euler

Ví dụ minh họa



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi
ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

23

Đường đi Euler

Đường đi Hamilton

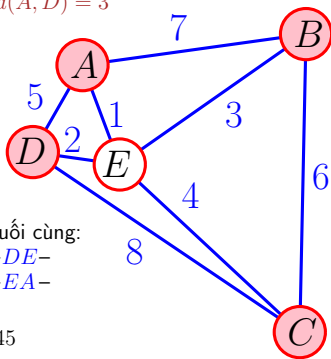
Tài liệu tham khảo

Các đỉnh bậc lẻ: A, B, C, D

$$d(A, B) = 4 \quad d(B, C) = 6 \quad d(C, D) = 6$$

$$d(A, C) = 5 \quad d(B, D) = 5$$

$$d(A, D) = 3$$



Chu trình trả lại cuối cùng:

$AB-BC-CD-DE-$

$EC-CB-BE-EA-$

$AD-DE-EA$

Độ dài đường đi: 45

Đường đi Hamilton

Ứng dụng: Bài toán người giao hàng (TSP)



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

24

■ Bài toán người giao hàng (Traveling Salesman Problem - TSP):

- Người bán hàng cần thăm n thành phố, mỗi thành phố đúng một lần, rồi quay về nơi xuất phát
- Mục tiêu: tìm hành trình có tổng chi phí (khoảng cách) nhỏ nhất

■ Mối liên hệ với đường đi Hamilton:

- Lời giải tối ưu của TSP chính là một chu trình Hamilton có trọng số nhỏ nhất
- Đồ thị mô hình hóa bài toán là đồ thị đầy đủ có trọng số trên các cạnh

■ Bài toán NP-khó: Không có thuật toán đa thức tìm nghiệm tối ưu (trừ khi $P = NP$)

■ Các ứng dụng trong thực tế:

- Tối ưu hóa lộ trình giao hàng
- Lập kế hoạch sản xuất
- Thiết kế mạch điện tử
- Định tuyến phương tiện
- Khoan đục bảng mạch (PCB)
- Quy hoạch tour du lịch
- Thu thập dữ liệu bằng drone

33

Đường đi Hamilton

Minh họa bài toán TSP



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi
ngắn nhất

Thuật toán Floyd

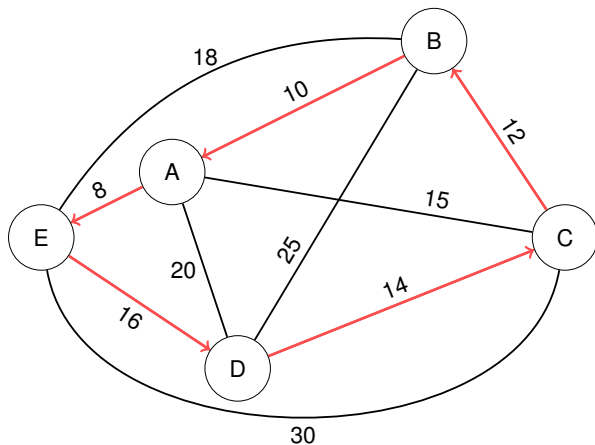
Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo



Hình: Ví dụ về bài toán TSP với 5 thành phố

Đường đi Hamilton

Ứng dụng: Mã Gray trong khoa học máy tính



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi
ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

26

- **Mã Gray (Gray Code):** Dãy các chuỗi nhị phân có độ dài n sao cho hai chuỗi liên tiếp chỉ khác nhau một bit
- **Mối liên hệ với đường đi Hamilton:**
 - Các mã Gray n -bit tương ứng với một đường đi Hamilton trong đồ thị siêu lập phương n chiều (n -cube)
 - Mỗi đỉnh là một chuỗi nhị phân độ dài n
 - Hai đỉnh kề nhau khi chuỗi nhị phân khác nhau đúng một bit
- **Ứng dụng trong lập trình và kỹ thuật số:**
 - Mã hóa vòng quay cho bộ mã hóa góc (encoder)
 - Tối thiểu hóa sai số trong chuyển đổi Analog-Digital
 - Tạo mã sửa lỗi và mã Hamming
 - Thuật toán backtracking và giải đệ quy

Đường đi Hamilton

Tạo mã Gray và đồ thị siêu lập phương

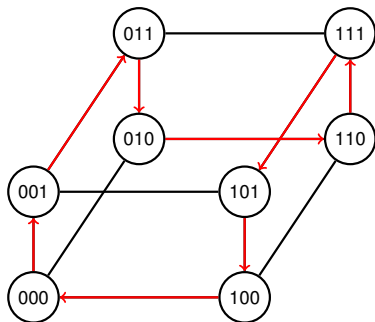


Mã Gray 3-bit:

- | | |
|-----------|-----------|
| ■ 000 (0) | ■ 111 (7) |
| ■ 001 (1) | ■ 101 (5) |
| ■ 011 (3) | ■ 100 (4) |
| ■ 010 (2) | → Quay về |
| ■ 110 (6) | 000 |

Tạo mã Gray đệ quy:

1. Bắt đầu với 0 và 1 (Mã Gray 1-bit)
2. Xây dựng mã Gray n -bit từ mã Gray $(n - 1)$ -bit
 - Sao chép danh sách mã Gray $(n - 1)$ -bit
 - Thêm 0 vào trước nửa đầu danh sách
 - Thêm 1 vào trước nửa sau danh sách



Hình: Khối 3 chiều Q_3 . Các đỉnh là các mã Gray 3-bit. Đường đi Hamilton ứng với các mã Gray 3-bit được đánh dấu bằng mũi tên đỏ

Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

27

33

Đường đi Hamilton

Ứng dụng của mã Gray trong kỹ thuật số



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi
ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

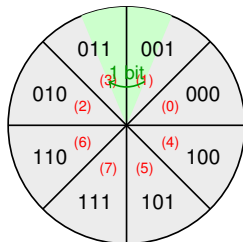
Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

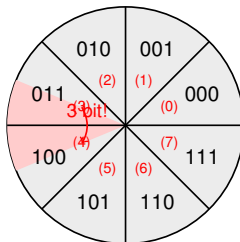
Tài liệu tham khảo

Mã Gray



001 → 011: chỉ 1 bit thay đổi

Mã nhị phân thông thường



011 → 100: tất cả 3 bit đều thay đổi

Ưu điểm của mã Gray: Chỉ một bit thay đổi giữa các vị trí kế nhau
⇒ Giảm thiểu lỗi đọc khi bộ mã hóa ở vị trí trung gian

Hình: So sánh bộ mã hóa vòng quay: Mã Gray (ít lỗi) và mã nhị phân thông thường (nhiều lỗi)

Đường đi Hamilton

Ứng dụng của mã Gray trong kỹ thuật số

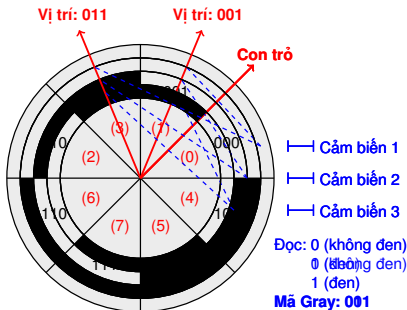


Bộ mã hóa vòng quay (rotary encoder):

- Chuyển đổi vị trí của con trỏ thành đầu ra số
- Sử dụng mã Gray để giảm thiểu lỗi khi đọc
- Khi sử dụng mã nhị phân thông thường, nhiều bit có thể thay đổi cùng một lúc

Ưu điểm của mã Gray:

- Chỉ một bit thay đổi mỗi lần
- Giảm thiểu sai số đọc (glitch)
- Tăng độ tin cậy của hệ thống



Hình: Bộ mã hóa vòng quay sử dụng mã Gray 3-bit

Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

29

33

Tài liệu tham khảo



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

30 Tài liệu tham khảo



Duan, Ran, Jiayi Mao, Xiao Mao, Xinkai Shu, and Longhui Yin (2025). “Breaking the Sorting Barrier for Directed Single-Source Shortest Paths”. In: *arXiv*: 2504.17033.



Lewis, Rhyd M. R. (2021). *Guide to Graph Colouring: Algorithms and Applications*. 2nd. Springer. DOI: 10.1007/978-3-030-81054-2.



Rosen, Kenneth (2012). *Discrete Mathematics and Its Applications*. 7th. McGraw-Hill.



Robertson, Neil, Daniel Sanders, Paul Seymour, and Robin Thomas (1997). “The four-colour theorem”. In: *Journal of Combinatorial Theory, Series B* 70.1, pp. 2–44. DOI: 10.1006/jctb.1997.1750.

Tài liệu tham khảo (tiếp)



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo



Appel, Kenneth and Wolfgang Haken (1977). “Every planar map is four colorable. Part I: Discharging”. In: *Illinois Journal of Mathematics* 21.3, pp. 429–490. DOI: 10.1215/ijm/1256049011.



Appel, Kenneth, Wolfgang Haken, and John Koch (1977). “Every planar map is four colorable. Part II: Reducibility”. In: *Illinois Journal of Mathematics* 21.3, pp. 491–567. DOI: 10.1215/ijm/1256049012.



Hart, Peter E., Nils J. Nilsson, and Bertram Raphael (1968). “A formal basis for the heuristic determination of minimum cost paths”. In: *IEEE Transactions on Systems Science and Cybernetics* 4.2, pp. 100–107. DOI: 10.1109/TSSC.1968.300136. This paper introduces the A* search algorithm.

Tài liệu tham khảo (tiếp)



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo

 Floyd, Robert W. (1962). “Algorithm 97: Shortest Path”. In: *Communications of the ACM* 5.6, p. 345. DOI: 10.1145/367766.368168.

 Warshall, Stephen (1962). “A theorem on Boolean matrices”. In: *Journal of the ACM* 9.1, pp. 11–12. DOI: 10.1145/321105.321107.

 Dijkstra, Edsger W. (1959). “A note on two problems in connexion with graphs”. In: *Numerische Mathematik* 1, pp. 269–271. DOI: 10.1007/BF01386390.

 Roy, Bernard (1959). “Transitivité et connexité”. In: *C. R. Acad. Sci. Paris* 249, pp. 216–218. (In French.) This paper introduced what later became known as the Roy-Warshall algorithm for transitive closure.

Tài liệu tham khảo (tiếp)



Lý thuyết đồ thị II

Hoàng Anh Đức

Bài toán đường đi
ngắn nhất

Thuật toán Floyd

Một số lỗi thường gặp

Một số ứng dụng

Đường đi Euler

Đường đi Hamilton

Tài liệu tham khảo



Fleury, M. (1883). “Deux problèmes de géométrie de situation”. In: *Journal de Mathématiques Élémentaires*, pp. 257–261. (In French.) This paper contains Fleury’s algorithm for finding Eulerian paths.



Hierholzer, Carl and Christian Wiener (1873). “Über die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren”. In: *Mathematische Annalen* 6.1, pp. 30–32. DOI: 10.1007/BF01442866. (In German.) This paper was published posthumously after Hierholzer’s death and edited by Christian Wiener.



Euler, Leonhard (1736). “Solutio problematis ad geometriam situs pertinentis”. In: *Commentarii Academiae Scientiarum Petropolitanae* 8, pp. 128–140. URL: <https://scholarlycommons.pacific.edu/euler-works/53/>. (In Latin.) English translation in *Scientific American*, 189 (1953), 66–70.

1. Logic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Đệ quy
4. Thuật toán I: Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp
5. Thuật toán II: Thuật toán tham lam, thuật toán đệ quy
6. Lý thuyết số cơ bản
7. Các phương pháp đếm
8. Lý thuyết đồ thị I: Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông
9. Lý thuyết đồ thị II: Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị
10. Lý thuyết đồ thị III: Cây
11. Đại số Boole

VNU-HUS MAT3500: Toán rời rạc

Lý thuyết đồ thị III

Cây

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn



Nội dung



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Giới thiệu

Một số tính chất của cây

Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

2 Giới thiệu

Một số tính chất của cây

Cây có gốc

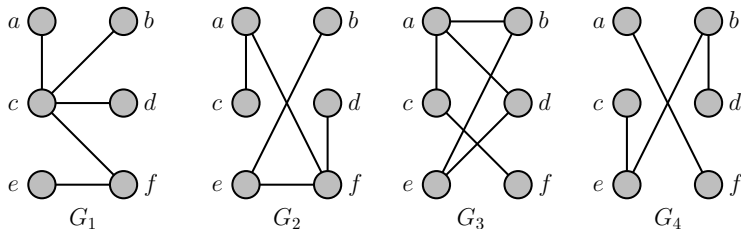
Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

- Đơn đồ thị vô hướng $T = (V, E)$ được gọi là một **cây (tree)** nếu T là đồ thị liên thông và không có chu trình
- Đơn đồ thị vô hướng $T = (V, E)$ được gọi là một **rừng (forest)** nếu T là đồ thị không có chu trình



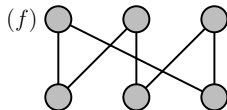
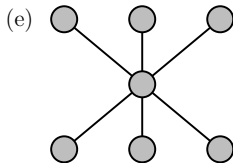
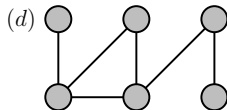
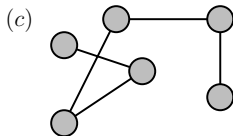
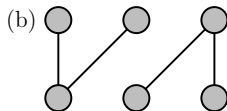
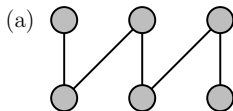
Hình: G_1, G_2 là cây. G_3, G_4 không là cây. G_4 là rừng. G_3 không là rừng

Bài tập 1

Các đồ thị hai phần đầy đủ $K_{m,n}$ ($m, n \in \mathbb{Z}^+$) nào là cây?

Bài tập 2

Đồ thị nào trong các đồ thị sau là cây?



3 Giới thiệu

Một số tính chất của cây

Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Cây

Một số tính chất của cây



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

4 Một số tính chất của cây

Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Định lý 1

Một đồ thị vô hướng G là một cây khi và chỉ khi giữa hai đỉnh bất kỳ của G tồn tại một đường đi đơn duy nhất

Chứng minh.

($\Rightarrow$) Giả sử G là cây

- Do G liên thông, với hai đỉnh bất kỳ $u, v \in V$, **tồn tại** một đường đi đơn giữa u và v
- Ta chứng minh đường đi này là **duy nhất**. Giả sử phản chứng rằng có ít nhất hai đường đi đơn $P = v_0, v_1, \dots, v_k$ và $Q = w_0, w_1, \dots, w_\ell$ khác nhau giữa hai đỉnh u và v , với $v_0 = w_0 = u$ và $v_k = w_\ell = v$
 - Do $v_0 = w_0$, tồn tại i thỏa mãn $v_j = w_j$ với mọi $0 \leq j \leq i$ và $v_{i+1} \neq w_{i+1}$
 - Do $v_k = w_\ell$, tồn tại $p \geq i + 1$ nhỏ nhất thỏa mãn $v_p \in \{w_i, \dots, w_\ell\}$ hoặc $w_p \in \{v_i, \dots, v_k\}$. Giả sử $v_p = w_q$ với $q \geq i$ nhỏ nhất có thể
 - $v_i, v_{i+1}, \dots, v_p$ và $w_q, \dots, w_{i+1}, w_i$ tạo thành một chu trình trong G , mâu thuẫn với giả thiết G là cây



Chứng minh (tiếp).

($\Leftarrow$) Giả sử với mọi cặp đỉnh $u, v \in V$, tồn tại một đường đi đơn duy nhất giữa u và v trong G

- Theo định nghĩa, G là *liên thông*
- Ta chứng minh G *không có chu trình*. Giả sử phản chứng rằng tồn tại một chu trình C trong G . Do đó, với hai đỉnh u, v bất kỳ thuộc C , có hai đường đi đơn khác nhau nối u và v , mâu thuẫn với giả thiết tồn tại duy nhất một đường đi đơn giữa hai đỉnh bất kỳ trong G



Cây

Một số tính chất của cây



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

6 Một số tính chất của cây

Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Định lý 2

Mọi cây $T = (V, E)$ với $|V| \geq 2$ có ít nhất hai đỉnh bậc 1

Chứng minh.

Lấy một đường đi đơn **dài nhất** $v_0, v_1, \dots, v_m$ trong T . Bậc của v_0 và v_m đều bằng 1, nếu không ta có thể tìm được một đường đi dài hơn $\square$

Định lý 3

Mọi cây gồm n đỉnh có chính xác $n - 1$ cạnh

Chứng minh.

Quy nạp theo n

- **Bước cơ sở:** Với $n = 1$, cây gồm 1 đỉnh có 0 cạnh
- **Bước quy nạp:** Giả sử mọi cây gồm k đỉnh có chính xác $k - 1$ cạnh, với $k \geq 1$. Ta chứng minh mọi cây T gồm $k + 1$ đỉnh có chính xác k cạnh. Thật vậy, do T là một cây, T có một đỉnh u bậc 1 nào đó. $T - u$ là một cây gồm k đỉnh và theo giả thiết quy nạp có $k - 1$ cạnh. Do đó, T có $(k - 1) + 1 = k$ cạnh $\square$



Bài tập 3

Chứng minh rằng một đồ thị $G = (V, E)$ là một cây khi và chỉ khi G không có chu trình và với mọi cặp đỉnh $u, v \in V$ thỏa mãn $uv \notin E$, đồ thị $G + uv$ có chính xác một chu trình

Bài tập 4

Giả sử một cây $T = (V, E)$ có số các cạnh là một số chẵn. Chứng minh rằng tồn tại một đỉnh trong T có bậc chẵn

Bài tập 5

Cho $T = (V, E)$ là một cây. Gọi Δ là bậc lớn nhất của một đỉnh trong T , nghĩa là, $\Delta = \max_{v \in V} \deg_T(v)$. Chứng minh rằng T có ít nhất Δ đỉnh bậc 1

Cây

Một số tính chất của cây



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

8

Một số tính chất của cây

Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Bài tập 6

Cho $T = (V, E)$ là một cây với $|V| > 1$. Giả sử nếu v là một đỉnh liền kề với một đỉnh bậc 1 trong T thì $\deg_T(v) \geq 3$. Chứng minh rằng tồn tại hai đỉnh bậc 1 trong T có chung một đỉnh liền kề với hai đỉnh đó

Bài tập 7

Một rừng bao gồm n đỉnh và t cây (mỗi cây là một thành phần liên thông của rừng) có bao nhiêu cạnh?

Cây

Cây có gốc



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

9 Cây có gốc

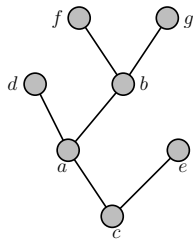
Duyệt cây có gốc

Cây bao trùm

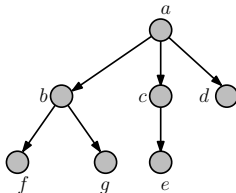
Cây bao trùm nhỏ nhất

Tài liệu tham khảo

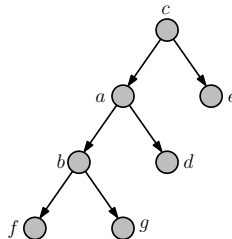
Một **cây có gốc (rooted tree)** là một cây trong đó một đỉnh được coi là **đỉnh gốc (root)** và mọi cạnh được định hướng xa khỏi đỉnh gốc. Ta ký hiệu một cây T với gốc r bằng cặp (T, r)



Cây T



Cây T với gốc a



Cây T với gốc c

Thông thường, một cây có gốc (T, r) được vẽ với **đỉnh gốc ở trên đỉnh của đồ thị**. Khi đó, ta có thể **bỏ qua các hướng của các cạnh**, do việc lựa chọn đỉnh gốc đã xác định các hướng này



- Với một đỉnh v khác đỉnh gốc của một cây có gốc T
 - **Đỉnh cha (parent)** của v là đỉnh u duy nhất sao cho có một cạnh có hướng từ u đến v . Ta cũng gọi v là **đỉnh con (child)** của u .
 - Đỉnh v được gọi là **đỉnh lá (leaf)** nếu v không có đỉnh con. Các đỉnh có đỉnh con được gọi là **các đỉnh trong (internal vertices)**
 - Các đỉnh có chung một đỉnh cha được gọi là **các đỉnh anh em (siblings)**
 - **Các đỉnh tổ tiên (ancestors)** của v là các đỉnh trên đường đi từ gốc tới đỉnh đó. **Các đỉnh con cháu (descendants)** của v là các đỉnh có v là một đỉnh tổ tiên
 - **Cây con (subtree) với gốc v** là đồ thị con của T cảm sinh bởi v và các đỉnh con cháu của nó
- Đỉnh gốc là đỉnh duy nhất trong cây **không có đỉnh cha** và là **tổ tiên của mọi đỉnh còn lại**. Đỉnh gốc luôn là một đỉnh trong, trừ trường hợp cây chỉ có một đỉnh duy nhất là đỉnh gốc thì đỉnh gốc là một đỉnh lá



- **Mức (level)** của một đỉnh trong một cây có gốc là độ dài của đường đi duy nhất từ gốc đến đỉnh đó. Mức của đỉnh gốc là 0
- **Độ cao (height)** h của một cây có gốc là độ dài lớn nhất của một đường đi từ gốc đến đỉnh lá. Nói cách khác, độ cao của một cây là mức lớn nhất của một đỉnh trong cây đó

Cây

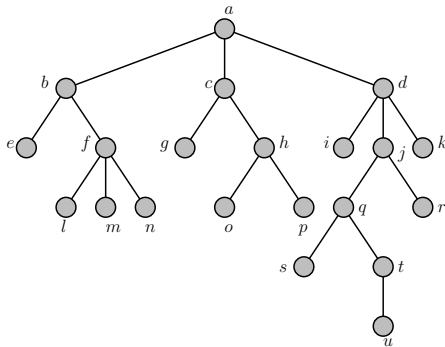
Cây có gốc



Bài tập 8

Hãy trả lời các câu hỏi về cây có gốc được minh họa như hình vẽ dưới đây

- (a) Đỉnh nào là đỉnh gốc?
- (b) Các đỉnh nào là đỉnh trong?
- (c) Các đỉnh nào là đỉnh lá?
- (d) Các đỉnh nào là đỉnh con của j ?
- (e) Đỉnh nào là đỉnh cha của h ?
- (f) Các đỉnh nào là đỉnh anh em của o ?
- (g) Các đỉnh nào là đỉnh tổ tiên của m ?



- (h) Các đỉnh nào là đỉnh con cháu của b ?
- (i) Mức của mỗi đỉnh là bao nhiêu?
- (j) Vẽ các cây con gốc c và gốc e

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

12 Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Cây

Cây có gốc



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

13 Cây có gốc

Duyệt cây có gốc

Cây bao trùm

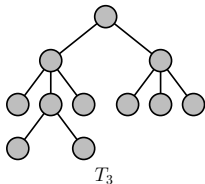
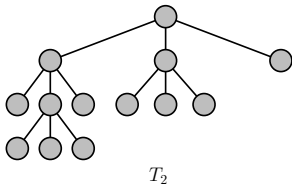
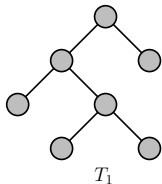
Cây bao trùm nhỏ nhất

Tài liệu tham khảo

- Một cây có gốc T được gọi là một **cây m -phân (m -ary tree)** ($m \geq 1$) nếu tất cả các đỉnh trong của T có tối đa là m đỉnh con. Nếu như mỗi đỉnh trong của T có chính xác m đỉnh con, ta gọi T là một **cây m -phân đầy đủ ($\text{full } m$ -ary tree)**
- Cây m -phân với $m = 2$ được gọi là **cây nhị phân (binary tree)**
- Một cây m -phân có độ cao h được gọi là **cân đối (balanced)** nếu tất cả các đỉnh lá có mức $h - 1$ hoặc h

Bài tập 9

Trong số các cây sau, với m là số nguyên dương nào đó, cây nào là cây m -phân? Cây nào là cây m -phân đầy đủ?





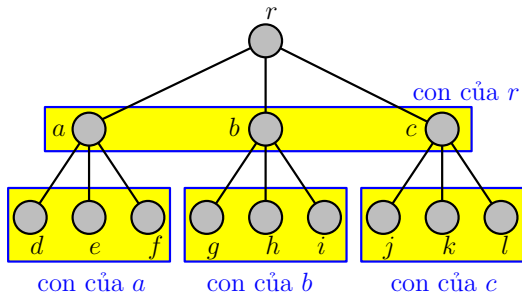
- Một *cây có gốc được sắp thứ tự (ordered rooted tree)* là một cây có gốc trong đó các đỉnh con của mỗi đỉnh được sắp xếp theo một thứ tự nào đó
- Ta thường vẽ cây có gốc được sắp thứ tự theo cách thông thường với hiểu ngầm rằng *các đỉnh con của mỗi đỉnh được sắp xếp theo thứ tự từ trái sang phải*
- Với một cây nhị phân được sắp thứ tự, nếu một đỉnh trong u có hai đỉnh con thì đỉnh thứ nhất được gọi là *đỉnh con trái (left child)*, đỉnh thứ hai được gọi là *đỉnh con phải (right child)*. Tương tự, cây con có gốc là đỉnh con trái được gọi là *cây con trái (left subtree)* và cây con có gốc là đỉnh con phải được gọi là *cây con phải (right subtree)* của u
- Các cây có gốc sắp thứ tự được ứng dụng rộng rãi trong khoa học máy tính, ví dụ như các cây phân tích cú pháp (parse trees), cấu trúc cây trong các tài liệu XML (XML documents), cấu trúc cây mô tả các thư mục và tệp (directories and file system), v.v...

Định lý 4

Mọi cây m -phân đầy đủ ($m \geq 1$) với i đỉnh trong có chính xác $n = m \cdot i + 1$ đỉnh

Chứng minh.

Mọi đỉnh khác đỉnh gốc là con của một đỉnh trong nào đó. Do đó có $m \cdot i$ đỉnh con (của các đỉnh trong) và 1 đỉnh gốc





Định lý 5

Một cây m -phân đầy đủ với

- (i) n đỉnh có $i = (n - 1)/m$ đỉnh trong và $\ell = ((m - 1)n + 1)/m$ đỉnh lá;
- (ii) i đỉnh trong có $n = mi + 1$ đỉnh và $\ell = (m - 1)i + 1$ đỉnh lá;
- (iii) ℓ đỉnh lá có $n = (m\ell - 1)/(m - 1)$ đỉnh và $i = (\ell - 1)/(m - 1)$ đỉnh trong

Bài tập 10

Sử dụng Định lý 4, hãy chứng minh Định lý 5

Bài tập 11

Chứng minh rằng với mọi $m \geq 1$, có nhiều nhất m^h đỉnh lá trong một cây m -phân có độ cao h . (**Gợi ý:** Quy nạp theo h)

Cây

Cây có gốc



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

17 Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Một ứng dụng của cây nhị phân là cấu trúc dữ liệu *cây tìm kiếm nhị phân (binary search tree)* sử dụng để *xây dựng các thuật toán tìm kiếm hiệu quả* khi *các phần tử trong tập hợp cần tìm kiếm đã được sắp xếp theo thứ tự nào đó*

Cây tìm kiếm nhị phân

- Là cây nhị phân: mỗi đỉnh có *tối đa 2 đỉnh con* (đỉnh con trái và đỉnh con phải)
- Mỗi đỉnh được gán nhãn bằng một *khóa (key)* tương ứng với phần tử trong tập hợp cần tìm kiếm
- Khóa ứng với một đỉnh *lớn hơn tất cả các khóa của đỉnh trong các cây con trái* và *nhỏ hơn tất cả các khóa của đỉnh trong các cây con phải*

Cây

Cây có gốc



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

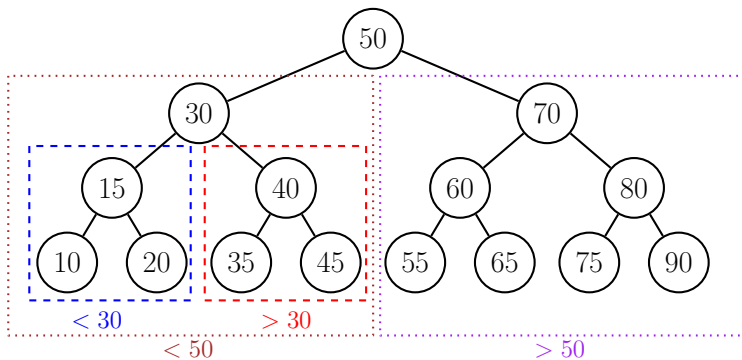
18 Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Hình: Ví dụ một cây tìm kiếm nhị phân

Thuật toán xây dựng cây tìm kiếm nhị phân

Đầu vào: Danh sách các phần tử $L = [k_1, k_2, \dots, k_n]$

Đầu ra: Cây tìm kiếm nhị phân T

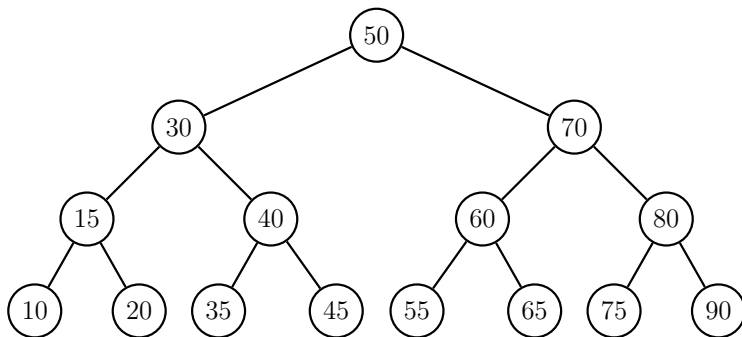
1. **Thủ tục** $\text{CreateBST}(L)$: (Tạo cây tìm kiếm nhị phân từ danh sách L)

- Khởi tạo T là cây rỗng với một đỉnh gốc r
- Gán $\text{key}(r) = k_1$
- Với mỗi i từ 2 đến n : Gọi thủ tục $\text{Insert}(T, r, k_i)$
- Trả về T

2. **Thủ tục** $\text{Insert}(T, v, k)$: (Chèn phần tử (có khóa bằng) k vào cây T với gốc v)

- Nếu $k < \text{key}(v)$:
 - Nếu v có con trái: Gọi đệ quy $\text{Insert}(T, \text{con trái của } v, k)$
 - Ngược lại: Tạo đỉnh mới w với $\text{key}(w) = k$ và thêm w làm con trái của v
- Nếu $k > \text{key}(v)$:
 - Nếu v có con phải: Gọi đệ quy $\text{Insert}(T, \text{con phải của } v, k)$
 - Ngược lại: Tạo đỉnh mới w với $\text{key}(w) = k$ và thêm w làm con phải của v

Ví dụ 1



Hình: Minh họa các bước xây dựng cây tìm kiếm nhị phân với danh sách các phần tử $L = [50, 30, 70, 15, 40, 60, 80, 10, 20, 35, 45, 55, 65, 75, 90]$

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

20 Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bài tập 12

Xây dựng cây tìm kiếm nhị phân với danh sách các phần tử:

(a) $L = [10, 15, 20, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 90]$

(b) $L = [50, 35, 75, 10, 60, 45, 20, 80, 55, 30, 90, 40, 70, 15, 65]$

Bài tập 13

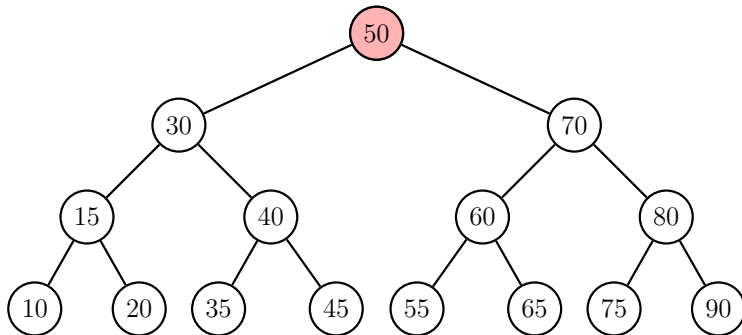
Sử dụng *thứ tự bảng chữ cái (alphabetical order)*¹, hãy xây dựng một cây tìm kiếm nhị phân cho các từ trong câu “The quick brown fox jumps over the lazy dog”

Bài tập 14

Từ thủ tục $\text{Insert}(T, v, k)$, hãy viết một thuật toán đệ quy tìm kiếm một phần tử có khóa bằng k trong cây tìm kiếm nhị phân T với gốc v và nếu không tìm được thì chèn một phần tử mới có khóa bằng k vào cây T và trả lại vị trí của phần tử mới này. (**Gợi ý:** Một phiên bản không đệ quy của thuật toán này là [Rosen 2012], Thuật toán 1, Trang 795)

¹ Thứ tự bảng chữ (alphabetical order) cái chính là thứ tự từ điển (lexicographic order) được áp dụng với các chữ cái

Ví dụ 2



$45 < 50$: đi sang cây con trái

Hình: Tìm kiếm phần tử có giá trị 45 trong cây tìm kiếm nhị phân

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

22 Cây có gốc

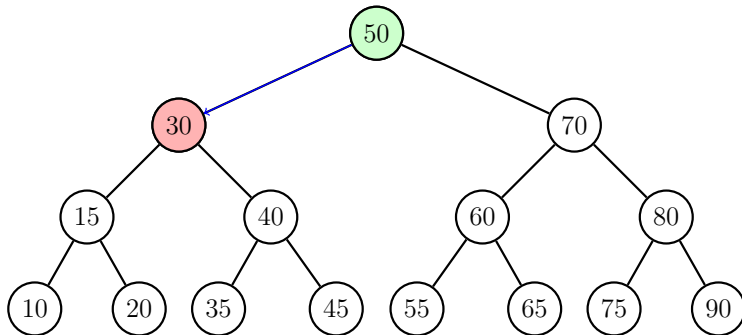
Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Ví dụ 2



$45 > 30$: đi sang cây con phải

Hình: Tìm kiếm phần tử có giá trị 45 trong cây tìm kiếm nhị phân

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

22 Cây có gốc

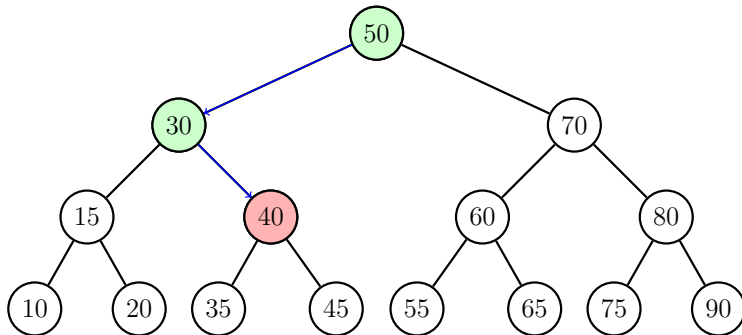
Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Ví dụ 2



$45 > 40$: đi sang cây con phải

Hình: Tìm kiếm phần tử có giá trị 45 trong cây tìm kiếm nhị phân

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của
cây

22 Cây có gốc

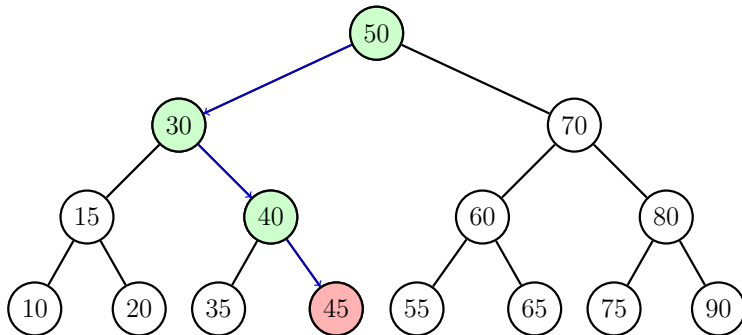
Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Ví dụ 2



45 = 45: Đã tìm thấy!

Hình: Tìm kiếm phần tử có giá trị 45 trong cây tìm kiếm nhị phân

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

22 Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Một ứng dụng khác của cây nhị phân được thể hiện trong định lý sau

Định lý 6

Mọi thuật toán sắp xếp n phần tử nào đó dựa trên các phép so sánh hai phần tử bất kỳ cần sử dụng ít nhất $\lceil \log(n!) \rceil$ phép so sánh trong trường hợp xấu nhất

- Do $\log(n!) = \Theta(n \log n)$, *một thuật toán sắp xếp n phần tử trong trường hợp xấu nhất cần $\Theta(n \log n)$ phép so sánh là một thuật toán tối ưu*, theo nghĩa là *không tồn tại thuật toán sắp xếp nào khác có thời gian chạy trong trường hợp xấu nhất tốt hơn*
- **Ý tưởng:** Mô hình các thuật toán bằng cây nhị phân đầy đủ với *các đỉnh trong là các phép so sánh* và *các đỉnh lá là các kết quả thu được* khi đi từ gốc (gọi là *cây quyết định (decision tree)*). *Số phép so sánh* sử dụng để ra một kết quả chính là *độ cao của cây quyết định*

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

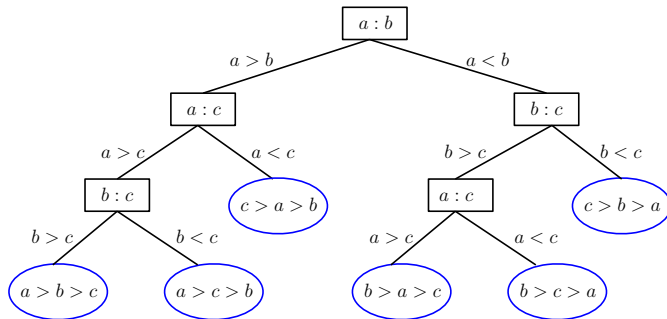
23 Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Hình: Mô hình thuật toán sắp xếp dãy ba phần tử a, b, c bằng cây quyết định

Bài tập 15

Với mọi $m \geq 1$, chứng minh rằng trong một cây m -phân có ℓ đỉnh lá và có độ cao h , ta có $h \geq \lceil \log_m \ell \rceil$

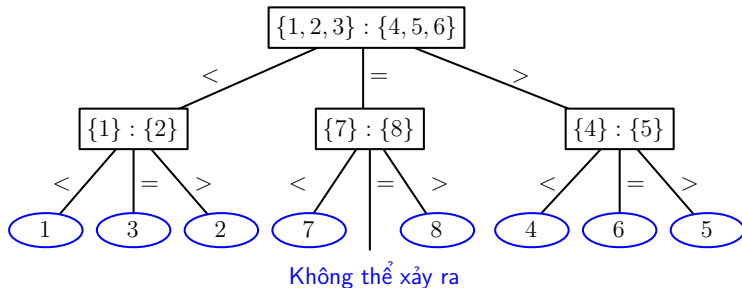


Ví dụ 3 (Ứng dụng khác của cây quyết định)

Ta sử dụng cây quyết định để giải câu đố tìm đồng xu giả quen thuộc sau: Giả sử có bảy đồng xu, tất cả có cùng trọng lượng, và một đồng xu giả có trọng lượng nhẹ hơn các đồng xu khác. Cần bao nhiêu lần cân bằng cách sử dụng một cân đĩa để xác định đồng xu nào là đồng xu giả? Hãy đưa ra thuật toán để tìm đồng xu giả này

- Một cân đĩa cho ra ba kết quả: bên trái nhẹ hơn, bên phải nhẹ hơn hoặc hai bên bằng nhau $\Rightarrow$ Cây quyết định là cây 3-phân
- Có tất cả 8 đồng xu $\Rightarrow$ Cây quyết định có ít nhất 8 đỉnh lá (mỗi đồng xu đều có thể là giả)
- Cây quyết định có độ cao $h \geq \lceil \log_3(8) \rceil = 2$ (Bài tập 15) $\Rightarrow$ Cần tối thiểu 2 lần cân để xác định đồng xu giả

Ví dụ 3 (tiếp)



Hình: Cây quyết định tìm đồng xu giả trong số 8 đồng xu đánh số 1, 2, ..., 8. Chỉ cần 2 lần cân là đủ để xác định đồng xu giả

Bài tập 16

Cần ít nhất bao nhiêu lần cân bằng cân đĩa để tìm một đồng xu giả trong số bốn đồng xu nếu đồng xu giả có thể nặng hơn hoặc nhẹ hơn các đồng xu khác? Mô tả một thuật toán để tìm đồng xu giả bằng số lần cân này

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

26 Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

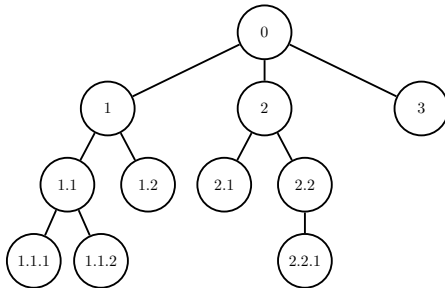


- Cây có gốc được sắp thứ tự thường được sử dụng để **lưu trữ thông tin**. Để truy cập dữ liệu, cần có các thuật toán để thăm (duyet) mỗi đỉnh của cây có gốc được sắp thứ tự
- Một số **thuật toán quan trọng** để **duyet tất cả các đỉnh của một cây có gốc được sắp thứ tự**
 - Duyệt tiền thứ tự (Preorder traversal)
 - Duyệt trung thứ tự (Inorder traversal)
 - Duyệt hậu thứ tự (Postorder traversal)
- Cây có gốc được sắp thứ tự cũng có thể được sử dụng để **biểu diễn các loại biểu thức khác nhau**, chẳng hạn như các biểu thức số học bao gồm số, biến và các phép toán. Thứ tự duyệt cây có gốc được sắp thứ tự cũng có thể được sử dụng trong việc **đánh giá (evaluation) các biểu thức biểu diễn bởi cây đó**



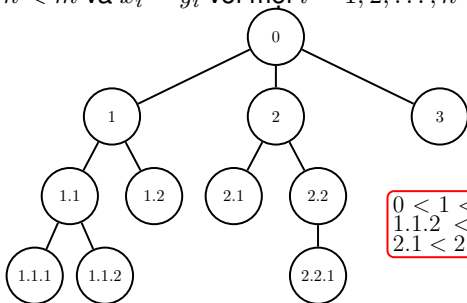
Hệ thống địa chỉ phổ quát (universal address system)

- Gốc có nhãn 0
- Các đỉnh con của gốc được gán nhãn $1, 2, \dots, k$ từ trái sang phải
- Với mỗi đỉnh v ở mức n với nhãn A , gán nhãn cho k_v đỉnh con của nó theo thứ tự từ trái sang phải, với các nhãn $A.1, A.2, \dots, A.k_v$



Trong hệ thống địa chỉ phổ quát:

- Với $n \geq 1$, đỉnh v ở mức n có nhãn $x_1 \dots x_n$, thể hiện đường đi duy nhất từ gốc đến v đi qua đỉnh x_1 ở mức 1, x_2 ở mức 2, $\dots$, x_n ở mức n
- Các đỉnh có thể được sắp xếp theo **thứ tự từ điển** (*lexicographic order*): Đỉnh có nhãn $x_1.x_2 \dots x_n$ nhỏ hơn đỉnh có nhãn $y_1.y_2 \dots y_m$ nếu tồn tại i , $0 \leq i \leq n$, sao cho $x_1 = y_1, x_2 = y_2, \dots, x_{i-1} = y_{i-1}$ và $x_i < y_i$; hoặc nếu $n < m$ và $x_i = y_i$ với mọi $i = 1, 2, \dots, n$



$0 < 1 < 1.1 < 1.1.1 < 1.1.2 < 1.2 < 2 < 2.1 < 2.2 < 2.2.1 < 3$



Bài tập 17

Giả sử địa chỉ của đỉnh v trong cây có gốc được sắp thứ tự T là 3.4.5.2.4.

- (a) v nằm ở mức nào?
- (b) Địa chỉ của đỉnh cha của v là gì?
- (c) Số lượng đỉnh anh em ít nhất mà v có thể có là bao nhiêu?
- (d) Nếu v có địa chỉ như trên, số lượng đỉnh nhỏ nhất có thể có trong T là bao nhiêu?
- (e) Tìm các địa chỉ khác mà bắt buộc phải xuất hiện trong cây.

Cây

Duyệt cây có gốc



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

Cây có gốc

31 Duyệt cây có gốc

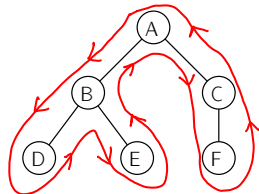
Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Duyệt theo *tiền thứ tự (preorder traversal)*, *trung thứ tự (inorder traversal)*, và *hậu thứ tự (postorder traversal)*

- Vẽ một *đường cong bao quanh cây có gốc được sắp thứ tự*, *bắt đầu từ gốc* và *di chuyển dọc theo các cạnh*
- Liệt kê các đỉnh theo *tiền thứ tự* bằng cách *liệt kê mỗi đỉnh ngay lần đầu tiên đường cong đi qua nó*
- Liệt kê các đỉnh theo *trung thứ tự* bằng cách *liệt kê một đỉnh lá ngay lần đầu tiên đường cong đi qua nó và liệt kê mỗi đỉnh trong vào lần thứ hai đường cong đi qua nó*
- Liệt kê các đỉnh theo *hậu thứ tự* bằng cách *liệt kê mỗi đỉnh vào lần cuối cùng nó được đường cong đi qua khi quay trở lại đỉnh cha của nó*



Tiền thứ tự: A, B, D, E, C, F

Trung thứ tự: D, B, E, A, F, C

Hậu thứ tự: D, E, B, F, C, A



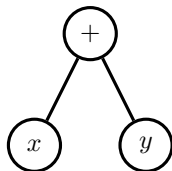
Có thể *biểu diễn các biểu thức phức tạp*, như các mệnh đề hợp thành, các phép kết hợp tập hợp, và các biểu thức số học bằng cách *sử dụng cây có gốc được sắp thứ tự*. Ví dụ, xét việc *biểu diễn một biểu thức số học (arithmetic expression)* liên quan đến các phép toán $+$ (cộng), $-$ (trừ), $*$ (nhân), $/$ (chia), và $\uparrow$ (lũy thừa)

- Các dấu ngoặc đơn được sử dụng để chỉ ra thứ tự của các phép toán
- Một cây có gốc được sắp thứ tự có thể được sử dụng để biểu diễn các biểu thức như vậy, trong đó các đỉnh trong biểu diễn các phép toán, và các đỉnh lá biểu diễn các biến hoặc số
- Mỗi phép toán thực hiện trên cây con trái và cây con phải của nó (theo thứ tự đó)



Ví dụ 4

1. Xây dựng cây con $x + y$

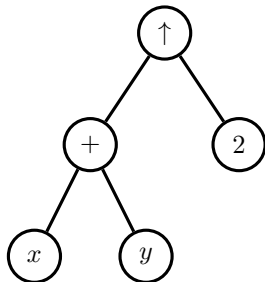


Hình: Xây dựng cây biểu diễn biểu thức $((x + y) \uparrow 2) + ((x - 4)/3)$



Ví dụ 4

2. Thêm phép lũy thừa $\uparrow 2$

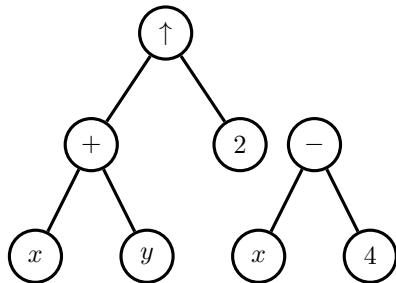


Hình: Xây dựng cây biểu diễn biểu thức $((x + y) \uparrow 2) + ((x - 4)/3)$



Ví dụ 4

3. Xây dựng cây con $x - 4$

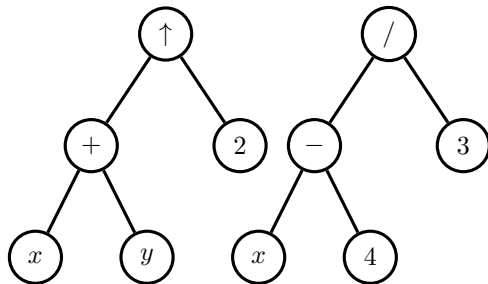


Hình: Xây dựng cây biểu diễn biểu thức $((x + y) \uparrow 2) + ((x - 4)/3)$



Ví dụ 4

4. Thêm phép chia cho 3

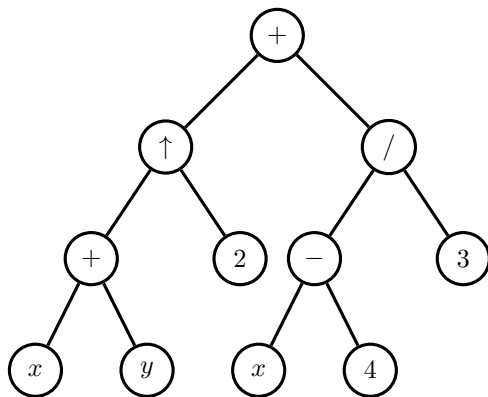


Hình: Xây dựng cây biểu diễn biểu thức $((x + y) \uparrow 2) + ((x - 4) / 3)$



Ví dụ 4

5. Kết hợp với gốc +

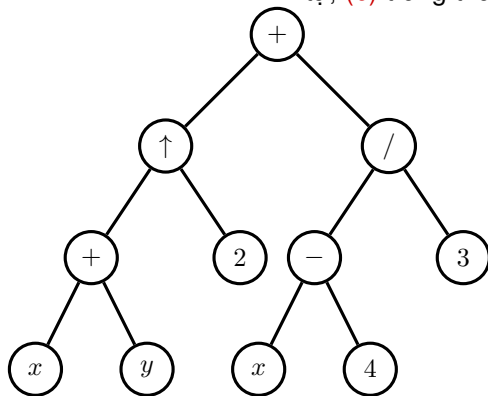


Hình: Xây dựng cây biểu diễn biểu thức $((x + y) \uparrow 2) + ((x - 4) / 3)$

Ví dụ 4

Bài tập 18

Liệt kê các đỉnh của cây trong Ví dụ 4 bằng cách duyệt theo (a) tiền thứ tự, (b) trung thứ tự, và (c) hậu thứ tự



Hình: Xây dựng cây biểu diễn biểu thức $((x + y) \uparrow 2) + ((x - 4) / 3)$

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

Cây có gốc

33 Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Cây

Duyệt cây có gốc



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

Cây có gốc

34 Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Cây nhị phân tương ứng với biểu thức số học (như trong Ví dụ 4) có thể được sử dụng để biểu diễn ***các dạng khác nhau của biểu thức*** đó dựa vào các cách duyệt cây

Dạng trung tố (infix form)

- Được tạo bằng cách duyệt cây theo ***trung thứ tự***: Toán hạng bên trái của toán tử được duyệt trước, sau đó là toán tử, và cuối cùng là toán hạng bên phải của toán tử
- ***Tính toán*** một biểu thức ở dạng trung tố:
 - Thực hiện ***từ trái sang phải***
 - Khi ***gặp một toán tử***, thực hiện phép toán tương ứng với ***hai toán hạng ngay bên trái và bên phải của toán tử này***
 - Khi ***một phép toán được thực hiện***, ***kết quả sẽ được xem như một toán hạng mới***

Ví dụ 5

- Dạng quen thuộc: $(x + y)^2 + \frac{x - 4}{3}$
- Đối với cây trong Ví dụ 4: $((x + y) \uparrow 2) + ((x - 4)/3)$ (cần thêm các ngoặc đơn mỗi khi gặp một toán tử để tránh nhầm lẫn)

Cây

Duyệt cây có gốc



Cây nhị phân tương ứng với biểu thức số học (như trong Ví dụ 4) có thể được sử dụng để biểu diễn *các dạng khác nhau của biểu thức* đó dựa vào các cách duyệt cây

Dạng tiền tố (prefix form)

- Được tạo bằng cách duyệt cây theo *tiền thứ tự*: Toán tử được duyệt trước các toán hạng của nó
- Còn được gọi là *ký pháp Ba Lan (Polish notation)* (đặt theo tên nhà logic học Ba Lan Jan Łukasiewicz)
- *Tính toán* một biểu thức ở dạng tiền tố:
 - Thực hiện *từ phải sang trái*
 - Khi *gặp một toán tử*, thực hiện phép toán tương ứng với *hai toán hạng ngay bên phải của toán tử này*
 - Khi *một phép toán được thực hiện*, *kết quả sẽ được xem như một toán hạng mới*

Ví dụ 6

- Đối với cây trong Ví dụ 4: $+ \uparrow + xy2 / - x43$ (không cần ngoặc đơn)

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

Cây có gốc

35 Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Cây nhị phân tương ứng với biểu thức số học (như trong Ví dụ 4) có thể được sử dụng để biểu diễn các dạng khác nhau của biểu thức đó dựa vào các cách duyệt cây

Dạng hậu tố (postfix form)

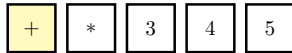
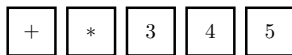
- Được tạo bằng cách duyệt cây theo *hậu thứ tự*: Toán tử được duyệt sau các toán hạng của nó
- Còn được gọi là ký pháp Ba Lan nghịch đảo (Reverse Polish notation)
- *Tính toán* một biểu thức ở dạng hậu tố:
 - Thực hiện *từ trái sang phải*
 - Khi *gặp một toán tử*, thực hiện phép toán tương ứng với *hai toán hạng ngay bên trái của toán tử này*
 - Khi *một phép toán được thực hiện*, *kết quả sẽ được xem như một toán hạng mới*

Ví dụ 7

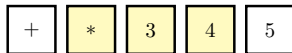
- Đối với cây trong Ví dụ 4: $xy + 2 \uparrow x4 - 3 / +$ (không cần ngoặc đơn)

Ví dụ 8 (Tính toán biểu thức dạng tiền tố)

Biểu thức tiền tố: $+ * 345$



Bước 1: Gặp toán tử $+$, cần hai toán hạng



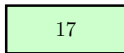
Bước 2: Tính $*34 = 3 \times 4 = 12$

$$3 \times 4 = 12$$



Bước 3: Tính $+125 = 12 + 5 = 17$

$$12 + 5 = 17$$



Kết quả cuối cùng

Hình: Minh họa quá trình tính toán biểu thức dạng tiền tố $+ * 345$

Ví dụ 9 (Tính toán biểu thức dạng hậu tố)

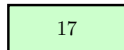
Biểu thức hậu tố: $3\ 4\ * \ 5\ +$



$$3 \times 4 = 12$$



$$12 + 5 = 17$$



Bước 1: Gặp $*$, áp dụng với hai toán hạng trước nó

Bước 2: Gặp $+$, áp dụng với hai toán hạng trước nó

Kết quả cuối cùng

Hình: Minh họa quá trình tính toán biểu thức dạng hậu tố $3\ 4\ * \ 5\ +$



Bài tập 19

- (a) Biểu diễn biểu thức $((x + 2) \uparrow 3) * (y - (3 + x)) - 5$ bằng cây nhị phân
- (b) Viết biểu thức này ở dạng tiền tố
- (c) Viết biểu thức này ở dạng hậu tố
- (d) Viết biểu thức này ở dạng trung tố

Bài tập 20

Vẽ cây có gốc được sắp thứ tự tương ứng với mỗi biểu thức số học được viết ở dạng tiền tố sau đây. Sau đó viết mỗi biểu thức ở dạng trung tố.

- (a) $+ * + - 53214$
- (b) $\uparrow +23 - 51$
- (c) $*/93 + *24 - 76$



Bài tập 21

Giá trị của mỗi biểu thức tiền tố sau là bao nhiêu?

(a) $- * 2/843$

(b) $\uparrow - * 33 * 425$

(c) $+ - \uparrow 32 \uparrow 23/6 - 42$

(d) $* + 3 + 3 \uparrow 3 + 333$

Bài tập 22

Giá trị của mỗi biểu thức hậu tố sau là bao nhiêu?

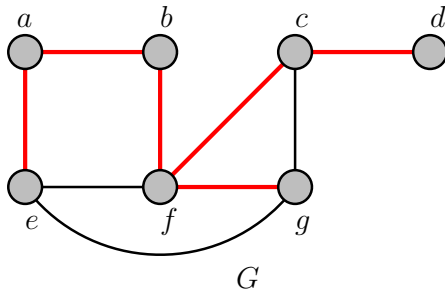
(a) $521 - -314 + +*$

(b) $93/5 + 72 - *$

(c) $32 * 2 \uparrow 53 - 84/ * -$



Cho $G = (V, E)$ là một đơn đồ thị. Một **cây bao trùm (spanning tree)** của G là một đồ thị con T của G thỏa mãn điều kiện T là một cây và T chứa tất cả các đỉnh của G





Định lý 7

Mọi đơn đồ thị liên thông G có một cây bao trùm

Chứng minh.

Ta xây dựng một cây bao trùm của G như sau

- Lấy một chu trình trong G nếu có
- Xóa một cạnh từ chu trình đó. Đồ thị mới thu được vẫn là liên thông
- Lặp lại các bước trên đến khi không có chu trình trong G

Do G có hữu hạn số đỉnh, cuối cùng ta thu được một cây bao trùm của G (Tại sao?) □

Bài tập 23

- Đơn đồ thị vô hướng liên thông nào có chính xác một cây bao trùm?
- Khi nào thì một cạnh của một đơn đồ thị vô hướng liên thông phải thuộc mọi cây bao trùm của đồ thị đó?

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

Cây có gốc

Duyệt cây có gốc

42 Cây bao trùm

Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Cho $G = (V, E)$ là một đơn đồ thị vô hướng liên thông. Ta trình bày thuật toán *duyệt theo chiều sâu (depth-first search, DFS)* để tìm một cây bao trùm của G

Thuật toán 1: Duyệt theo chiều sâu

Input: G : đơn đồ thị vô hướng liên thông với các đỉnh

$v_1, v_2, \dots, v_n$

Output: Cây bao trùm T

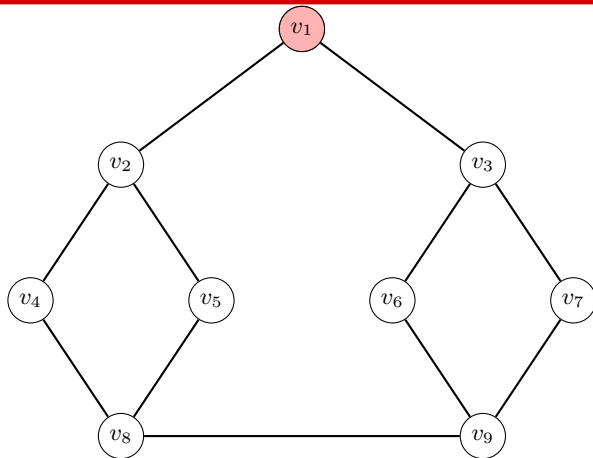
1 **procedure** dfs(G):

2 $T :=$ cây có chính xác một đỉnh v_1
3 visit(v_1)

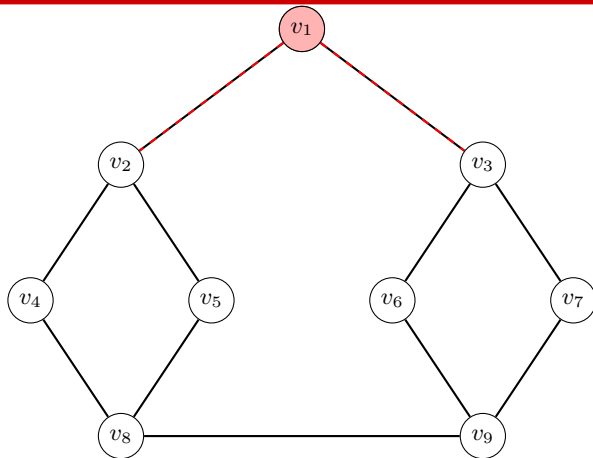
4 **procedure** visit(v : một đỉnh của G):

5 **for** mỗi đỉnh w liền kề với v và chưa thuộc T **do**
6 Thêm w và cạnh vw vào T
7 visit(w)

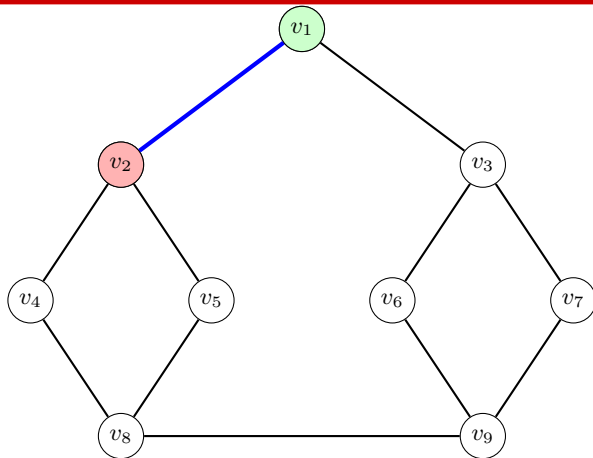
8 **return** T



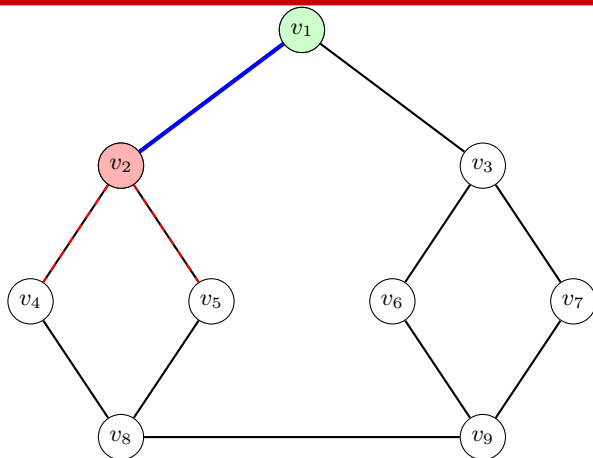
Bắt đầu thuật toán: Khởi tạo T với đỉnh v_1



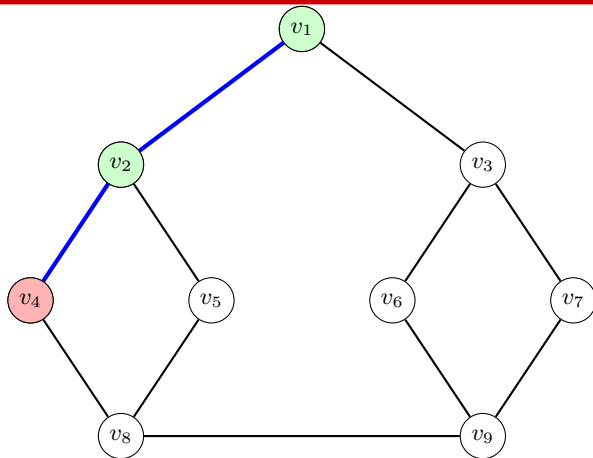
Gọi $\text{visit}(v_1)$, xem xét các đỉnh liền kề v_2 và v_3



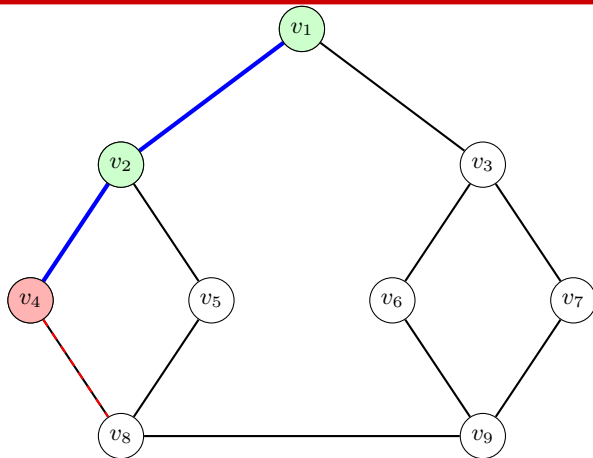
Thêm v_2 và cạnh v_1v_2 vào T



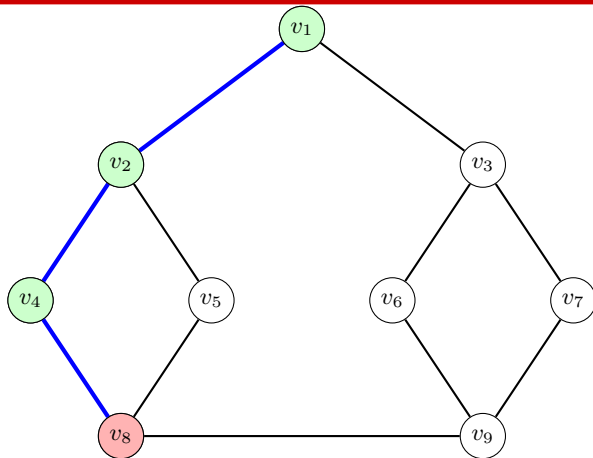
Gọi $\text{visit}(v_2)$, xem xét các đỉnh liền kề v_4 và v_5



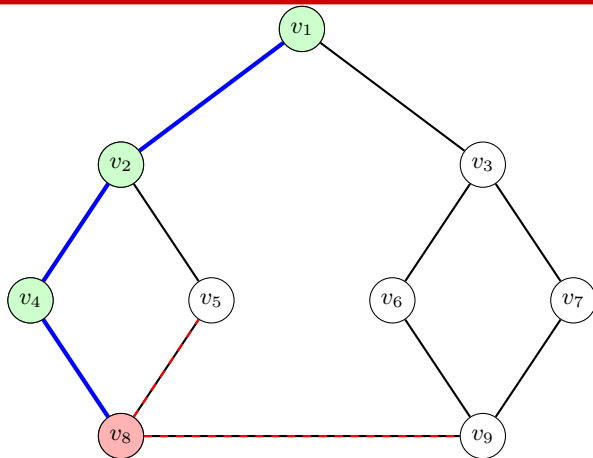
Thêm v_4 và cạnh v_2v_4 vào T



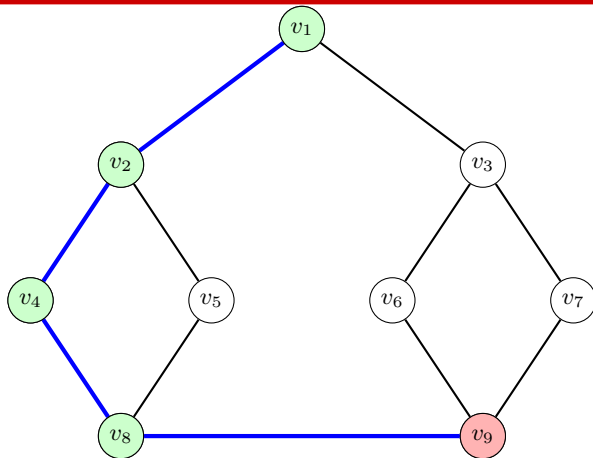
Gọi $\text{visit}(v_4)$, xem xét đỉnh liền kề v_8



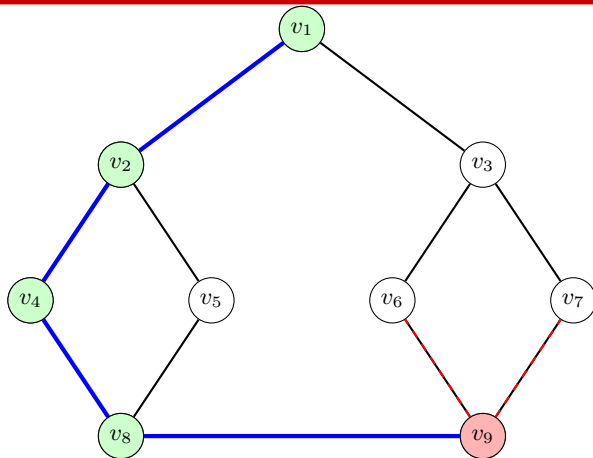
Thêm v_8 và cạnh v_4v_8 vào T



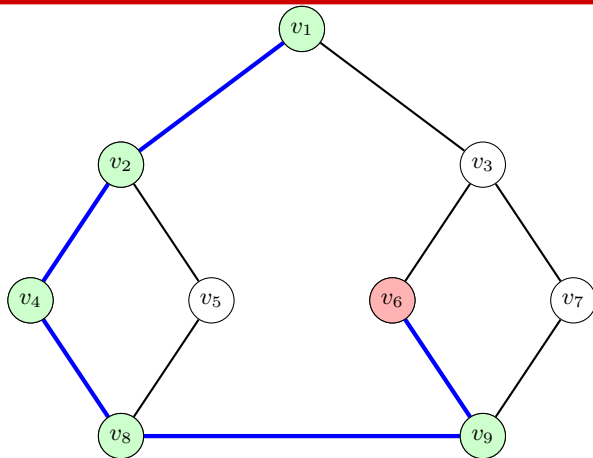
Gọi $\text{visit}(v_8)$, xem xét đỉnh liền kề v_5 và v_9



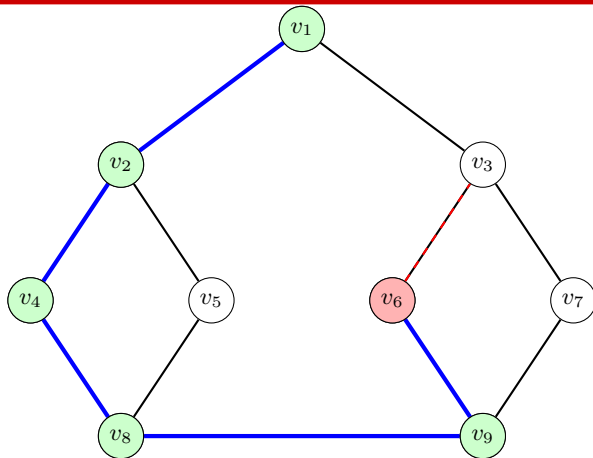
Thêm v_9 và cạnh v_8v_9 vào T



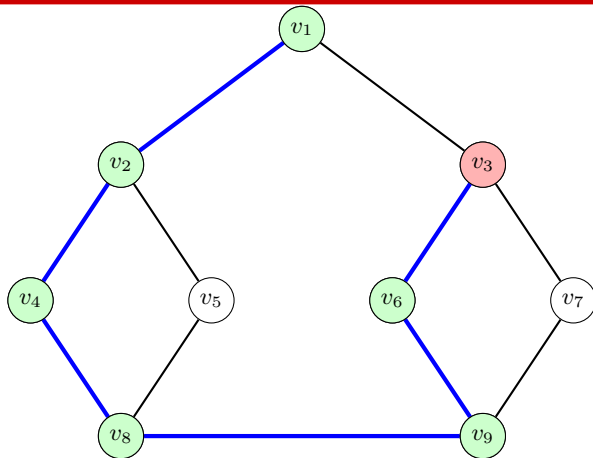
Gọi $\text{visit}(v_9)$, xem xét đỉnh liền kề v_6 và v_7 chưa thuộc T



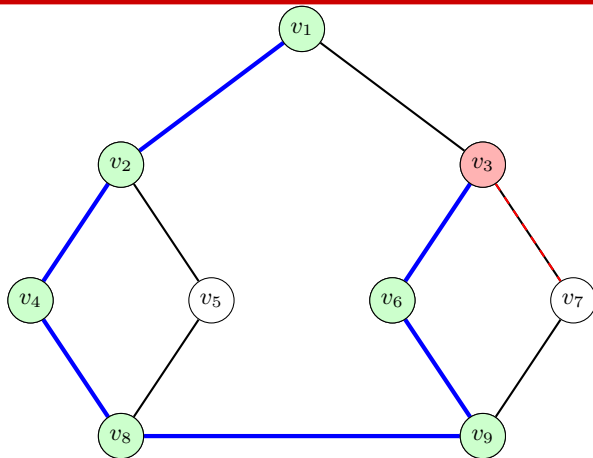
Thêm v_6 và cạnh v_9v_6 vào T



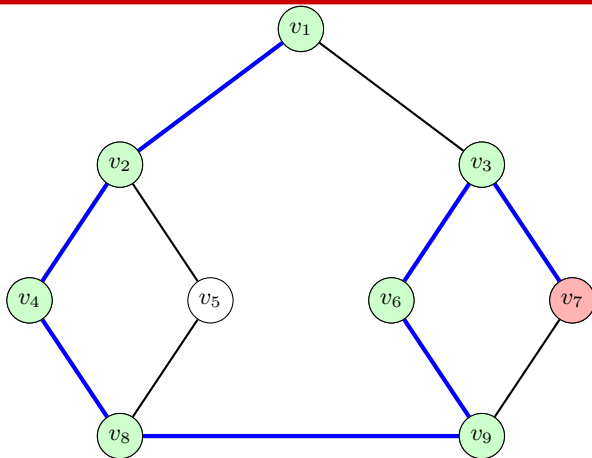
Gọi $\text{visit}(v_6)$, xem xét đỉnh liền kề v_3



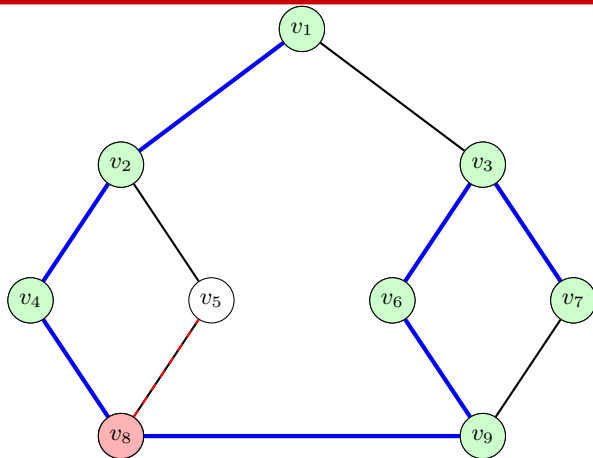
Thêm v_3 và cạnh v_6v_3 vào T



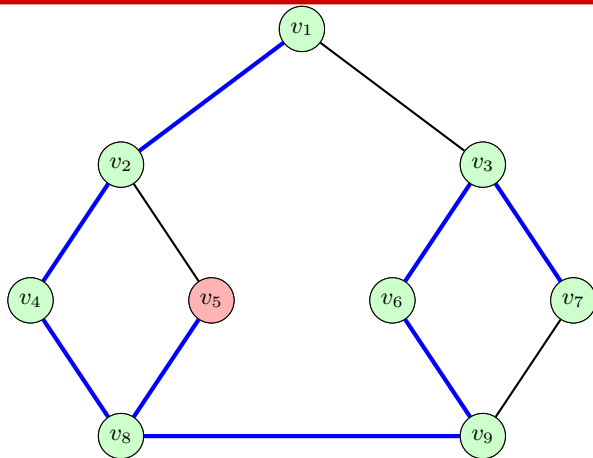
Gọi $\text{visit}(v_3)$, xem xét đỉnh liền kề v_7 chưa thuộc T



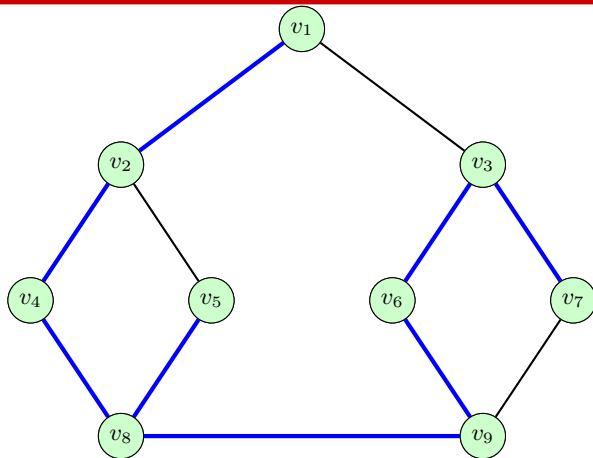
Thêm v_7 và cạnh v_3v_7 vào T



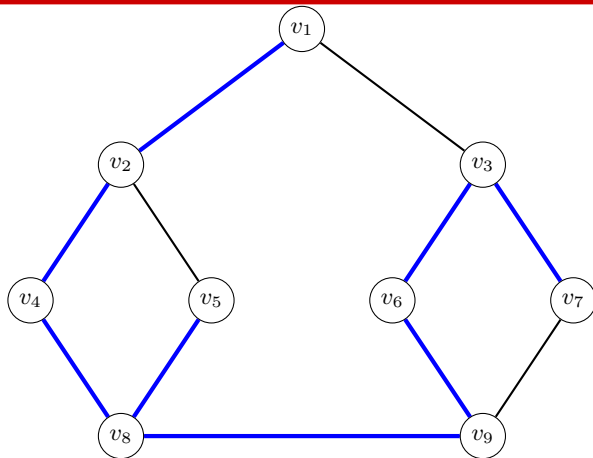
Quay lui đến $\text{visit}(v_8)$, xét đỉnh v_5 chưa thuộc T



Thêm v_5 và cạnh v_8v_5 vào T



Thuật toán hoàn thành: Cây bao trùm T được tìm thấy



Thứ tự duyệt: $v_1, v_2, v_4, v_8, v_9, v_6, v_3, v_7, v_5$

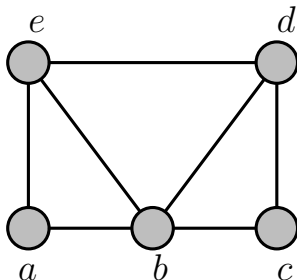


- Trước mỗi lần lặp trong thủ tục `visit()`, T là cây chứa toàn bộ các đỉnh đã được gọi bởi thủ tục `visit()`
- Thuật toán DFS chạy trong thời gian $O(n^2)$ với đơn đồ thị vô hướng liên thông $G = (V, E)$ bất kỳ, trong đó $n = |V|$
 - `visit()` được gọi cho mỗi đỉnh chính xác một lần
 - Một cạnh e trong G được xét tối đa hai lần để xác định xem e và một đầu mút của e có thể được thêm vào cây T hay không
 - Do đó, thuật toán DFS chạy trong thời gian $O(|E|)$. Chú ý rằng $|E| \leq n(n-1)/2 = O(n^2)$
- Thuật toán DFS có thể được sử dụng như là cơ sở để giải quyết nhiều bài toán khác, ví dụ như bài toán tìm các đường đi và chu trình trong đồ thị, bài toán xác định các thành phần liên thông, bài toán tìm đỉnh cắt, v.v... DFS cũng là cơ sở cho *kỹ thuật quay lui (backtracking technique)* được sử dụng để thiết kế các giải thuật cho nhiều bài toán khó



- Có những bài toán chỉ có thể giải được bằng cách tìm kiếm trên tập toàn bộ các lời giải. Để tiến hành tìm kiếm lời giải một cách có hệ thống, một hướng tiếp cận là sử dụng cây quyết định với mỗi đỉnh trong mô tả một quyết định và mỗi đỉnh lá ứng với một lời giải nào đó
- Trong *kỹ thuật quay lui (backtracking)*, để tìm kiếm lời giải mong muốn, đầu tiên ta tiến hành một dãy các quyết định sao cho ta có thể tìm đến các lời giải ở càng xa đỉnh gốc càng tốt. Một dãy các quyết định này tương ứng với một đường đi trong cây quyết định từ gốc đến lá
- Một khi ta biết rằng không thể tìm đến một lời giải mới xa hơn thông qua dãy các quyết định, ta lui lại xét đỉnh cha của đỉnh đang xét hiện tại, và tiếp tục tìm đến một lời giải khác dựa trên một dãy các quyết định mới nếu có thể
- Quá trình này được lặp lại cho đến khi tìm được lời giải mong muốn hoặc kết luận rằng không có lời giải nào như thế tồn tại

Bài tập 24 (★)



Hãy tô màu đồ thị trên bằng 3 màu đỏ (**R**), xanh lá cây (**G**), và xanh da trời (**B**) dựa trên kỹ thuật quay lui đã mô tả. (**Gợi ý:** Xem [Rosen 2012], Mục 11.4.4)

Cho $G = (V, E)$ là một đơn đồ thị vô hướng liên thông. Ta trình bày thuật toán *duyệt theo chiều rộng (breadth-first search, BFS)* để tìm một cây bao trùm của G

Thuật toán 2: Duyệt theo chiều rộng

Input: G : đơn đồ thị vô hướng liên thông với các đỉnh $v_1, v_2, \dots, v_n$

Output: Cây bao trùm T

```
1  $T :=$  cây chỉ chứa duy nhất một đỉnh  $v_1$ 
2  $L :=$  danh sách rỗng
3 Thêm  $v_1$  vào danh sách  $L$  các đỉnh chưa xét
4 while  $L$  khác rỗng do
5     |   Bỏ đi đỉnh thứ nhất  $v$  từ  $L$ 
6     |   for mỗi đỉnh  $w$  liền kề với  $v$  do
7     |   |   if  $w$  không thuộc  $L$  và  $w$  không thuộc  $T$  then
8     |   |   |   Thêm  $w$  vào cuối danh sách  $L$ 
9     |   |   |   Thêm  $w$  và cạnh  $vw$  vào  $T$ 
10 return  $T$ 
```

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

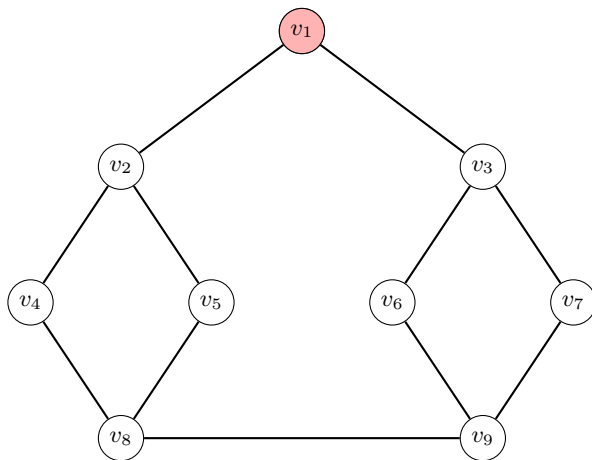
Cây có gốc

Duyệt cây có gốc

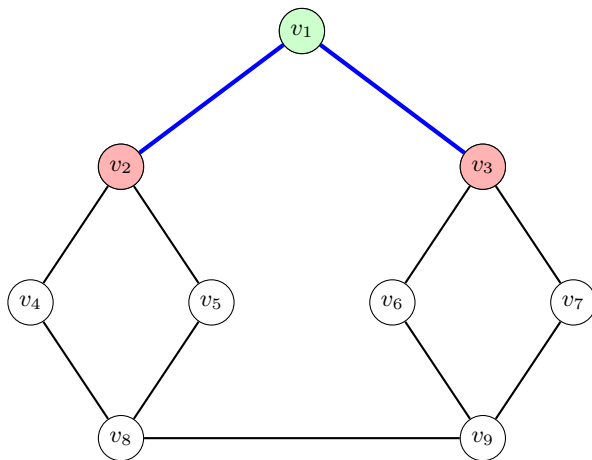
48 Cây bao trùm

Cây bao trùm nhỏ nhất

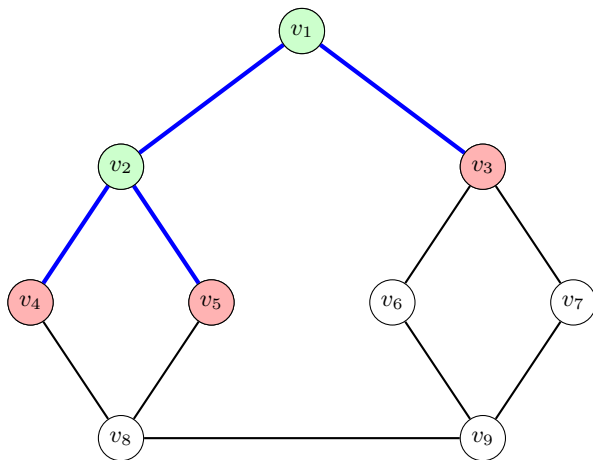
Tài liệu tham khảo



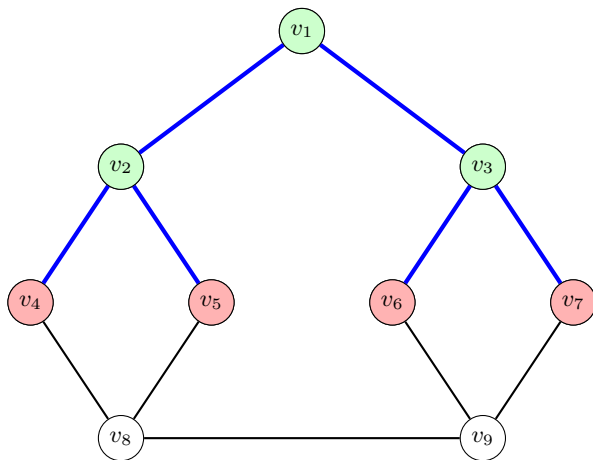
Danh sách $L = [v_1]$



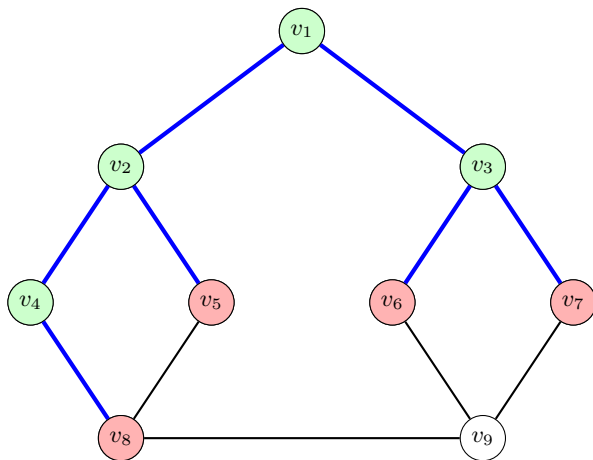
Danh sách $L = [v_2, v_3]$



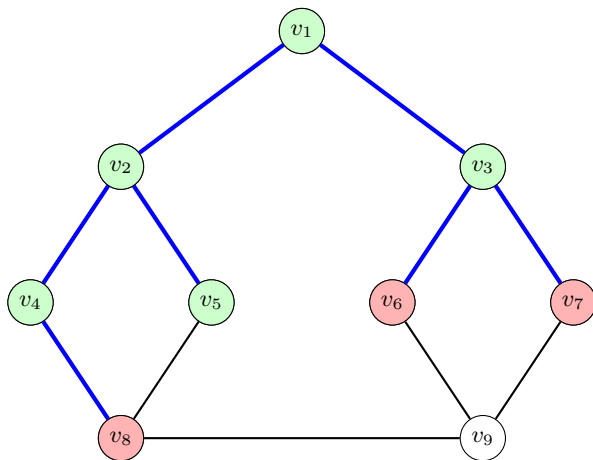
Danh sách $L = [v_3, v_4, v_5]$



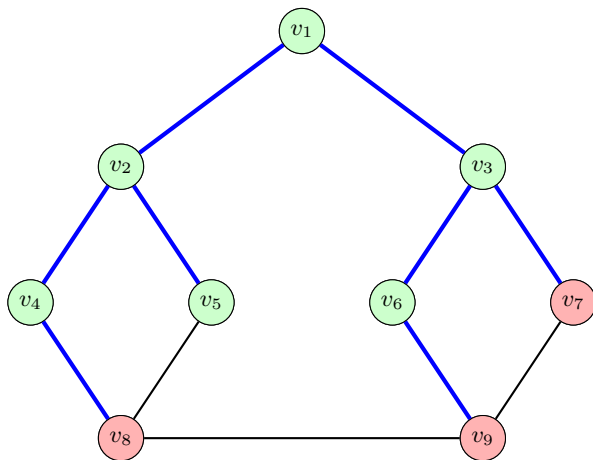
Danh sách $L = [v_4, v_5, v_6, v_7]$



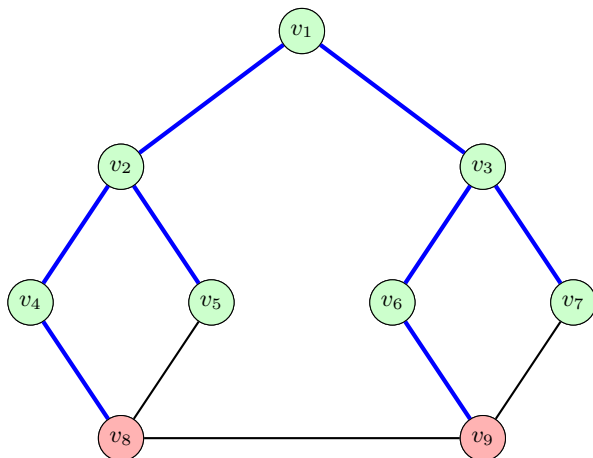
Danh sách $L = [v_5, v_6, v_7, v_8]$



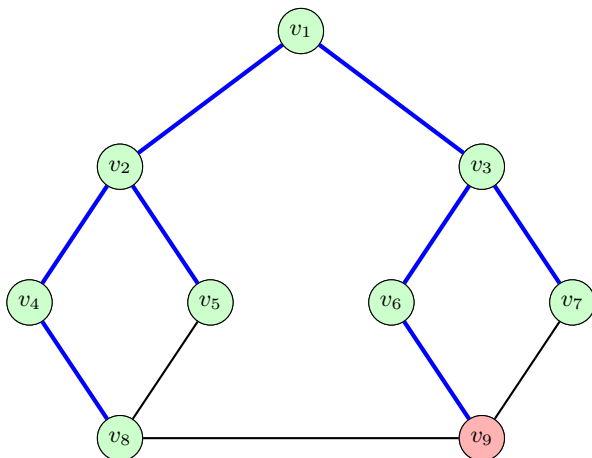
Danh sách $L = [v_6, v_7, v_8]$



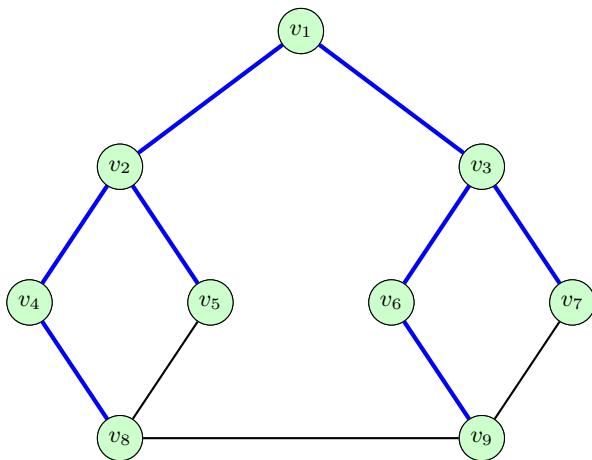
Danh sách $L = [v_7, v_8, v_9]$



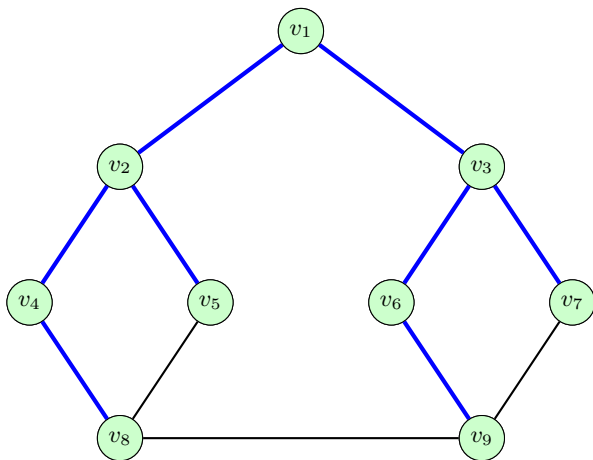
Danh sách $L = [v_8, v_9]$



Danh sách $L = [v_9]$



Danh sách $L = []$



Thứ tự thăm các đỉnh: $v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9$

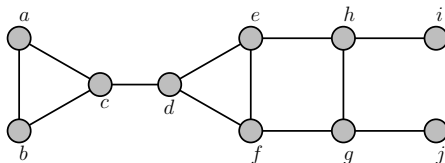


- Trước mỗi lần lặp vòng **while**, T là cây chứa các đỉnh đã xét (= các đỉnh đã bỏ đi từ danh sách L)
- Thuật toán BFS chạy trong thời gian $O(n^2)$ với đơn đồ thị vô hướng liên thông $G = (V, E)$ bất kỳ, trong đó $n = |V|$
 - Với mỗi đỉnh v , thuật toán xét các đỉnh liền kề với v và thêm các đỉnh chưa xét vào cuối danh sách L
 - Mỗi cạnh được xét nhiều nhất hai lần để xác định xem có cần thêm cạnh đó và một đỉnh đầu mút vào cây T hay không
 - Do đó, thuật toán BFS chạy trong thời gian $O(|E|)$, hay nói cách khác $O(n^2)$
- Thuật toán BFS là cơ sở để thiết kế các thuật toán giải nhiều bài toán khác nhau, ví dụ như bài toán tìm các thành phần liên thông, bài toán xác định xem một đồ thị có phải đồ thị hai phần hay không, bài toán tìm đường đi ngắn nhất giữa hai đỉnh, v.v...

Bài tập 25

Tìm một cây bao trùm của đồ thị sau bằng

- (1) thuật toán tìm kiếm theo chiều sâu (DFS);
 - (2) thuật toán tìm kiếm theo chiều rộng (BFS);
- bắt đầu từ đỉnh a



Bài tập 26

Hãy mô tả các cây bao trùm xuất ra bởi thuật toán tìm kiếm theo chiều rộng và thuật toán tìm kiếm theo chiều sâu khi chạy trên đồ thị đầy đủ K_n , với n là số nguyên dương nào đó

Cây

Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

Cây có gốc

Duyệt cây có gốc

Cây bao trùm

52 Cây bao trùm nhỏ nhất

Tài liệu tham khảo

Cho $G = (V, E, w)$ là một đồ thị liên thông vô hướng có trọng số. Một **cây bao trùm nhỏ nhất (minimum spanning tree)** của G là một cây bao trùm của G có tổng trọng số các cạnh của cây là nhỏ nhất có thể

- **Input:** Đồ thị liên thông vô hướng có trọng số $G = (V, E, w)$
- **Output:** Một cây bao trùm nhỏ nhất T của G

- Thuật toán Prim
- Thuật toán Kruskal



Thuật toán 3: Thuật toán Prim

Input: $G = (V, E, w)$: đồ thị liên thông vô hướng có trọng số

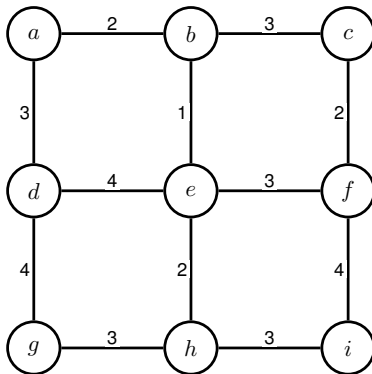
Output: Một cây bao trùm nhỏ nhất T của G

- 1 $T :=$ một cạnh có trọng số nhỏ nhất
 - 2 **for** $i := 1$ **to** $n - 2$ **do**
 - 3 $e :=$ một cạnh có trọng số nhỏ nhất liên thuộc với một đỉnh của T thỏa mãn $T + e$ không có chu trình
 - 4 $T := T + e$
 - 5 **return** T
-



Ví dụ 10

Tìm một cây bao trùm nhỏ nhất T của đồ thị $G = (V, E, w)$ sau bằng thuật toán Prim



G

Cây

Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

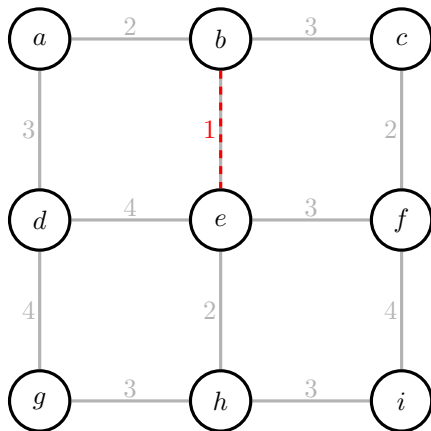
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

55 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bắt đầu: Chọn đỉnh b làm đỉnh khởi đầu

Cây

Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

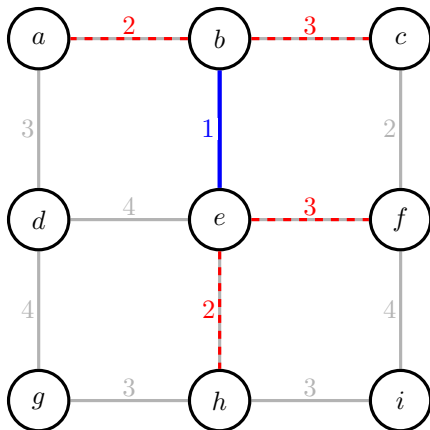
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

55 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 1: Thêm cạnh (b, e) vào T . Tổng trọng số = 1

Cây

Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

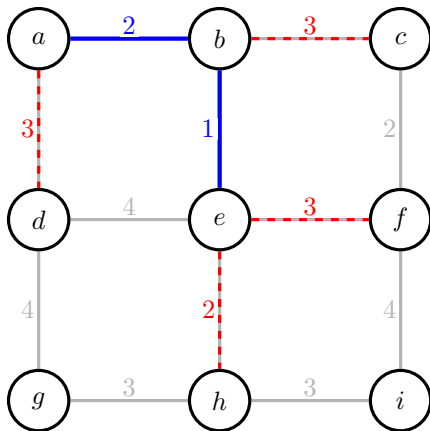
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

55 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 2: Thêm cạnh (b, a) vào T . Tổng trọng số = 3

Cây

Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

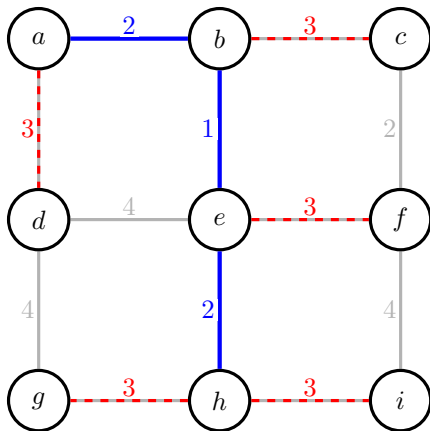
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

55 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 3: Thêm cạnh (e, h) vào T . Tổng trọng số = 5

Cây

Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

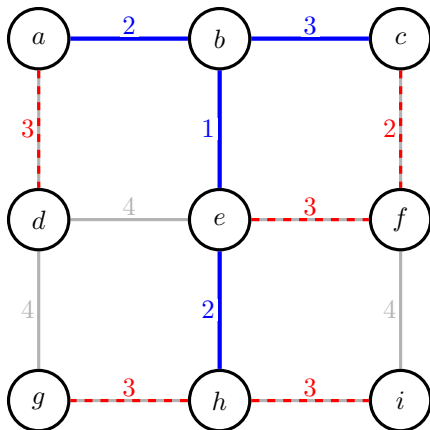
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

55 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 4: Thêm cạnh (b, c) vào T . Tổng trọng số = 8

Cây

Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

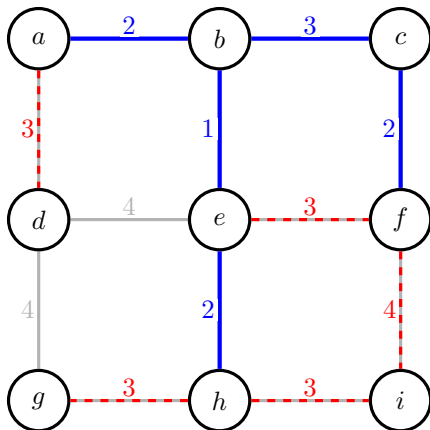
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

55 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 5: Thêm cạnh (c, f) vào T . Tổng trọng số = 10

Cây

Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

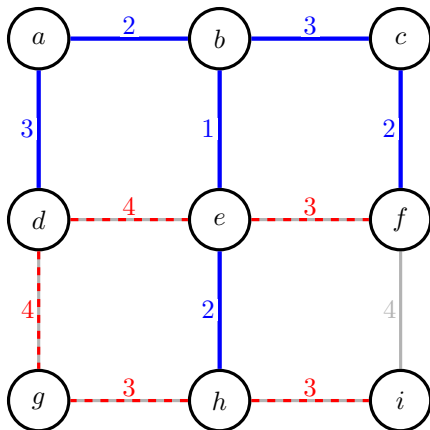
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

55 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 6: Thêm cạnh (a, d) vào T . Tổng trọng số = 13

Cây

Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

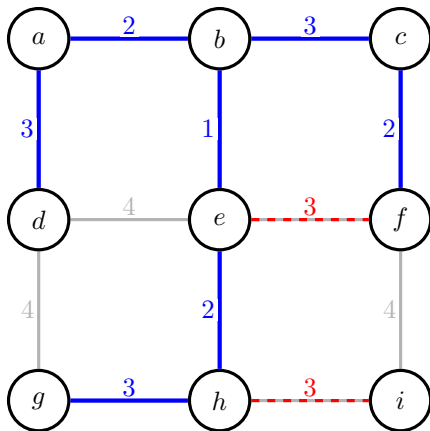
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

55 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 7: Thêm cạnh (h, g) vào T . Tổng trọng số = 16

Cây

Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

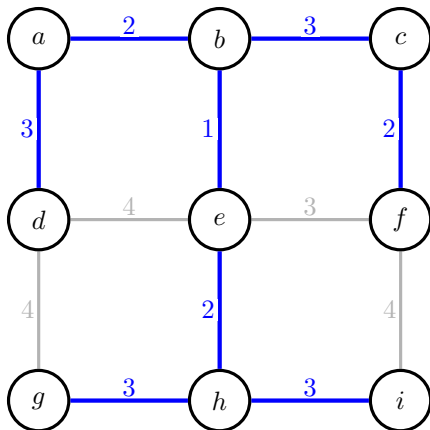
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

55 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 8: Thêm cạnh (h, i) vào T . Tổng trọng số = 19

Cây

Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

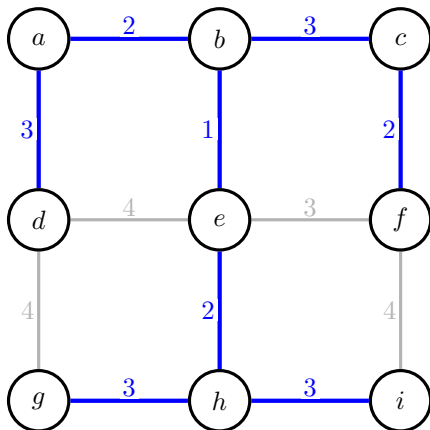
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

55 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Kết quả: Cây bao trùm nhỏ nhất với tổng trọng số = 19



Định lý 8

Nếu T là một cây bao trùm xuất ra bởi thuật toán Prim với đồ thị đầu vào $G = (V, E, w)$ thì T là một cây bao trùm nhỏ nhất của G

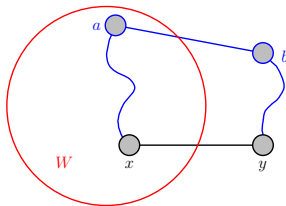
Chứng minh.

- Thuật toán Prim luôn xuất ra một cây bao trùm
 - Ở các bước trung gian, T luôn là cây
 - Mỗi bước trung gian, thuật toán thêm một cạnh và một đỉnh mới vào T
- Ta chứng minh bằng phản chứng. Giả sử T không có tổng trọng số nhỏ nhất
- Gọi $ET = (e_1, e_2, \dots, e_m)$ là dãy các cạnh được chọn theo thứ tự thực hiện của thuật toán Prim
- Gọi U là một cây bao trùm nhỏ nhất thỏa mãn điều kiện U có chứa k cạnh đầu tiên của ET với k lớn nhất có thể, nghĩa là $e_1, e_2, \dots, e_k$ thuộc U và $e_{k+1} = xy$ không thuộc U



Chứng minh (tiếp).

- Gọi W là tập các đỉnh của T tính đến thời điểm *ngay trước khi* $e_{k+1} = xy$ *được chọn*
- Gọi P là một đường đi giữa x và y trong U (chú ý rằng xy không là một cạnh của U) và gọi ab là cạnh đầu tiên của P có một đầu mút (a) thuộc W và một đầu mút (b) không thuộc W
- Xét cây bao trùm $S = U - ab + xy$
 - Nếu $w(a, b) > w(x, y)$: Tổng trọng số của S nhỏ hơn của U , mâu thuẫn với giả thiết U là cây bao trùm nhỏ nhất
 - Nếu $w(a, b) = w(x, y)$: S là một cây bao trùm nhỏ nhất và S có chứa nhiều cạnh của ET hơn U , mâu thuẫn với định nghĩa của U
 - Nếu $w(a, b) < w(x, y)$: Thuật toán Prim sẽ chọn cạnh ab thay vì cạnh xy , mâu thuẫn với giả thiết ban đầu là xy là cạnh được chọn





- Thuật toán Prim là một thuật toán tham lam
 - Ở mỗi bước xây dựng cây bao trùm nhỏ nhất T , thuật toán Prim chỉ thêm các cạnh có trọng số nhỏ nhất liên kết với một đỉnh của cây T tính đến hiện tại mà không tạo thành chu trình mới
- Thuật toán Prim có thể được lập trình để chạy trong thời gian $O(m \log n)$ với đồ thị $G = (V, E, w)$ thỏa mãn $m = |E|$ và $n = |V|$
- Thuật toán Prim tương tự như thuật toán Dijkstra tìm đường đi ngắn nhất giữa hai đỉnh đã giới thiệu trước đó



Thuật toán 4: Thuật toán Kruskal

Input: $G = (V, E, w)$: đồ thị liên thông vô hướng có trọng số

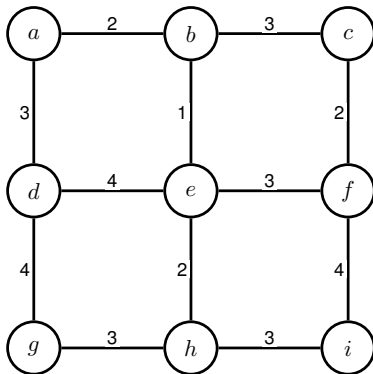
Output: Một cây bao trùm nhỏ nhất T của G

- ```
1 $T :=$ đồ thị rỗng (không có đỉnh và cạnh)
2 for $i := 1$ to $n - 1$ do
3 $e :=$ một cạnh có trọng số nhỏ nhất thỏa mãn $T + e$
 không có chu trình
4 $T := T + e$
5 return T
```
-



### Ví dụ 11

Tìm một cây bao trùm nhỏ nhất  $T$  của đồ thị  $G = (V, E, w)$  sau bằng thuật toán Kruskal



$G$

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

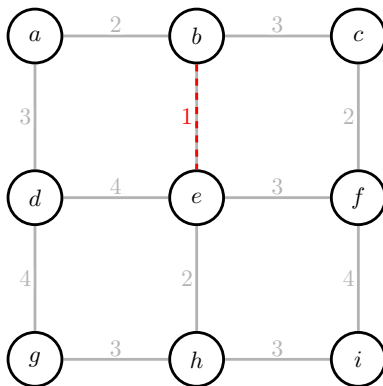
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bắt đầu:  $T$  là đồ thị rỗng. Xét các cạnh theo thứ tự trọng số tăng dần.

**Hình:** Minh họa các bước của thuật toán Kruskal

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

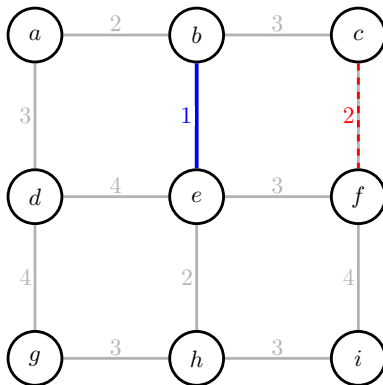
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 1: Thêm cạnh  $(b, e)$  vào  $T$ . Tổng trọng số = 1.

**Hình:** Minh họa các bước của thuật toán Kruskal

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

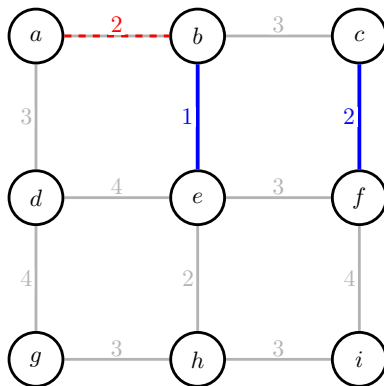
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 2: Thêm cạnh  $(c, f)$  vào  $T$ . Tổng trọng số = 3.

**Hình:** Minh họa các bước của thuật toán Kruskal

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

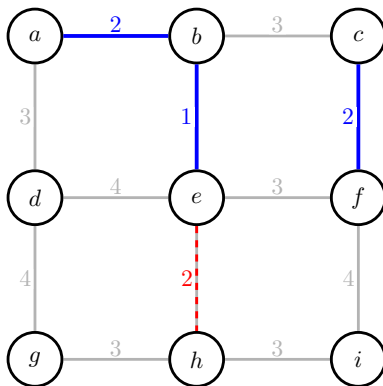
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 3: Thêm cạnh  $(a, b)$  vào  $T$ . Tổng trọng số = 5.

**Hình:** Minh họa các bước của thuật toán Kruskal

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

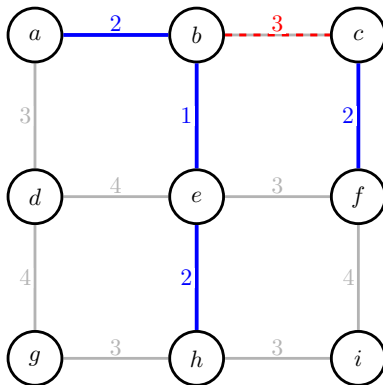
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 4: Thêm cạnh  $(e, h)$  vào  $T$ . Tổng trọng số = 7.

**Hình:** Minh họa các bước của thuật toán Kruskal

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

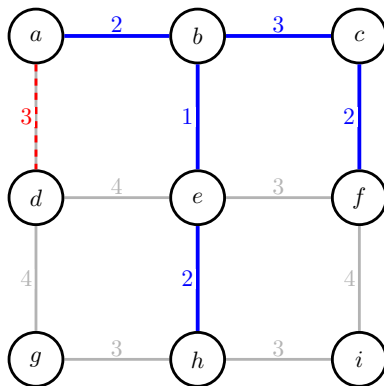
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 5: Thêm cạnh  $(b, c)$  vào  $T$ . Tổng trọng số = 10.

**Hình:** Minh họa các bước của thuật toán Kruskal

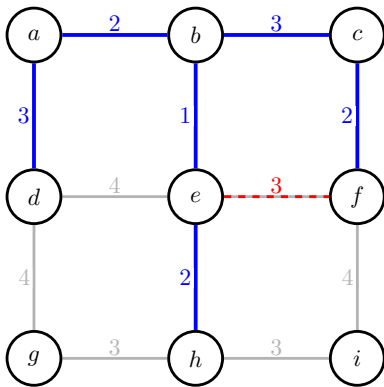
### Cây bao trùm nhỏ nhất



Hoàng Anh Đức

61) Cây bao trùm nhỏ nhất

## Tài liệu tham khảo



Bước 6: Thêm cạnh  $(a, d)$  vào  $T$ . Tổng trọng số = 13.

**Hình:** Minh họa các bước của thuật toán Kruskal

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

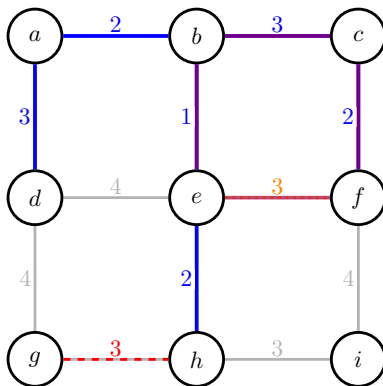
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 7: Cạnh  $(e, f)$  bị loại vì tạo chu trình. Xét cạnh tiếp theo.

**Hình:** Minh họa các bước của thuật toán Kruskal

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

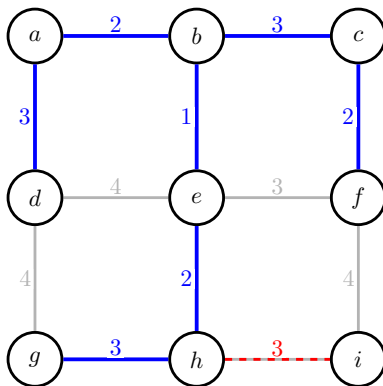
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 8: Thêm cạnh  $(g, h)$  vào  $T$ . Tổng trọng số = 16.

**Hình:** Minh họa các bước của thuật toán Kruskal

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

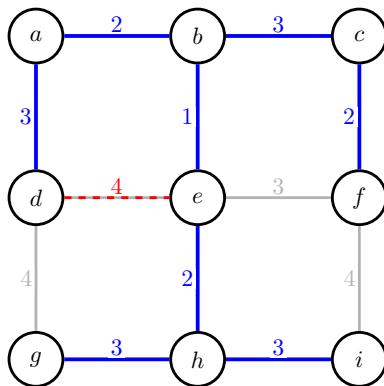
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 9: Thêm cạnh  $(h, i)$  vào  $T$ . Tổng trọng số = 19.

**Hình:** Minh họa các bước của thuật toán Kruskal

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

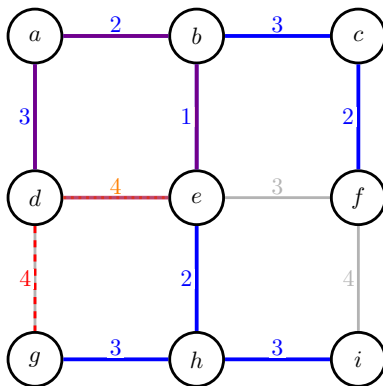
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 10: Cạnh  $(d, e)$  bị loại vì tạo chu trình. Xét cạnh tiếp theo.

**Hình:** Minh họa các bước của thuật toán Kruskal

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

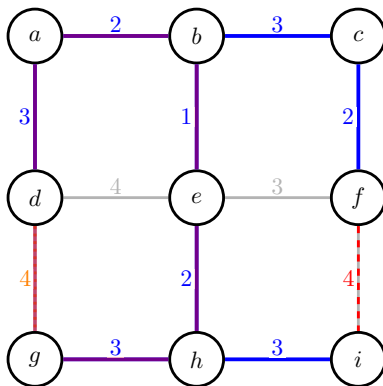
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 11: Cạnh  $(d, g)$  bị loại vì tạo chu trình. Xét cạnh tiếp theo.

**Hình:** Minh họa các bước của thuật toán Kruskal

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

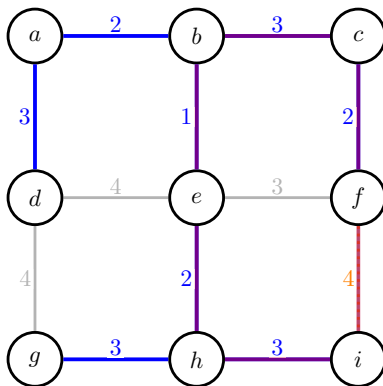
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Bước 12: Cạnh  $(f, i)$  bị loại vì tạo chu trình.

**Hình:** Minh họa các bước của thuật toán Kruskal

# Cây

## Cây bao trùm nhỏ nhất



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

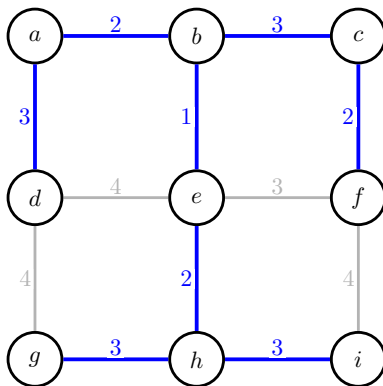
Cây có gốc

Duyệt cây có gốc

Cây bao trùm

61 Cây bao trùm nhỏ nhất

Tài liệu tham khảo



Kết quả: Cây bao trùm nhỏ nhất với tổng trọng số = 19

Hình: Minh họa các bước của thuật toán Kruskal



## Bài tập 27

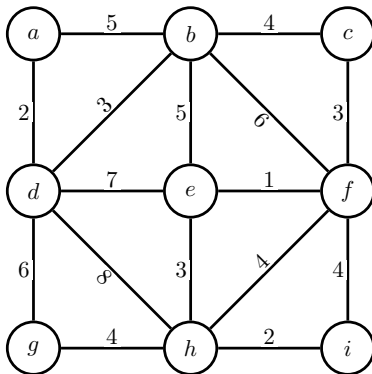
Chứng minh rằng nếu  $T$  là một cây bao trùm của  $G = (V, E, w)$  xuất ra bởi thuật toán Kruskal thì  $T$  là cây bao trùm nhỏ nhất (**Gợi ý:** xem lại chứng minh tính đúng đắn của thuật toán Prim)

- Thuật toán Kruskal là một thuật toán tham lam
  - Ở mỗi bước xây dựng cây bao trùm nhỏ nhất  $T$ , thuật toán Kruskal chỉ thêm các cạnh có trọng số nhỏ nhất mà không tạo thành chu trình mới
  - Khác với thuật toán Prim, *ở các bước trung gian của thuật toán Kruskal,  $T$  có thể không là cây*
- Thuật toán Kruskal có thể được lập trình để chạy trong thời gian  $O(m \log m)$  với đồ thị  $G = (V, E, w)$  thỏa mãn  $m = |E|$



### Bài tập 28

Tìm một cây bao trùm nhỏ nhất của đồ thị sau



bằng cách sử dụng

(a) thuật toán Prim

(b) thuật toán Kruskal

Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của cây

Cây có gốc

Duyệt cây có gốc

Cây bao trùm

63 Cây bao trùm nhỏ nhất

Tài liệu tham khảo

# Tài liệu tham khảo



Lý thuyết đồ thị III

Hoàng Anh Đức

Giới thiệu

Một số tính chất của  
cây

Cây có gốc

Duyệt cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

64 Tài liệu tham khảo



Rosen, Kenneth (2012). *Discrete Mathematics and Its Applications*. 7th. McGraw-Hill.

1. Logic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Đệ quy
4. Thuật toán I: Mô tả, chứng minh, đánh giá thuật toán; Tìm kiếm và sắp xếp
5. Thuật toán II: Thuật toán tham lam, thuật toán đệ quy
6. Lý thuyết số cơ bản
7. Các phương pháp đếm
8. Lý thuyết đồ thị I: Giới thiệu, Biểu diễn đồ thị và sự đẳng cấu, Tính liên thông
9. Lý thuyết đồ thị II: Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị
10. Lý thuyết đồ thị III: Cây
11. Đại số Boole

# VNU-HUS MAT3500: Toán rời rạc

## Đại số Boole

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học  
Đại học KHTN, ĐHQG Hà Nội  
[hoanganhduc@hus.edu.vn](mailto:hoanganhduc@hus.edu.vn)



# Nội dung



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

# Hàm Boole



Đại số Boole  
Hoàng Anh Đức

## 2 Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

- **Đại số Boole (Boolean Algebra)** đưa ra các phép toán và quy tắc làm việc với tập  $\{0, 1\}$  (ứng với  $\{F, T\}$ )
- Các phép toán được sử dụng nhiều nhất trong Đại số Boole
  - **Phần bù (complement):**  $\overline{0} = 1$  và  $\overline{1} = 0$  (ứng với  $\neg$ )
  - **Tổng (sum):**  $0 + 0 = 0$ ,  $0 + 1 = 1$ ,  $1 + 0 = 1$ , và  $1 + 1 = 1$  (ứng với  $\vee$ )
  - **Tích (product):**  $0 \cdot 0 = 0$ ,  $0 \cdot 1 = 0$ ,  $1 \cdot 0 = 0$ , và  $1 \cdot 1 = 1$  (ứng với  $\wedge$ )

## Bài tập 1

Tính  $1 \cdot 0 + \overline{(0 + 1)}$

## Bài tập 2

Dịch tương đương logic  $(T \wedge T) \vee \neg F \equiv T$  sang một đẳng thức trong Đại số Boole.

# Hàm Boole



Đại số Boole  
Hoàng Anh Đức

Giả sử  $B = \{0, 1\}$

- $B^n = \{(x_1, x_2, \dots, x_n) \mid x_i \in B \text{ với } 1 \leq i \leq n\}$
- Một biến  $x$  được gọi là một **biến Boole (Boolean variable)** nếu nó chỉ nhận các giá trị từ  $B$
- Một hàm  $F : B^n \rightarrow B$  được gọi là một **hàm Boole bậc  $n$  (Boolean function of degree  $n$ )**. Các giá trị của một hàm Boole thường được cho bởi các bảng
- Các **biểu thức Boole (Boolean expression)** với các biến  $x_1, x_2, \dots, x_n$  được định nghĩa một cách đệ quy như sau:
  - $0, 1, x_1, x_2, \dots, x_n$  là các biểu thức Boole
  - Nếu  $E$  là một biểu thức Boole thì  $\overline{E}$  cũng là
  - Nếu  $E_1$  và  $E_2$  là các biểu thức Boole thì  $(E_1 + E_2)$ , và  $(E_1 \cdot E_2)$  cũng là
- **Mỗi biểu thức Boole biểu diễn một hàm Boole**. Các giá trị của hàm này nhận được bằng cách thay 0 và 1 cho các biến trong biểu thức đó
- Hàm Boole  $F$  bậc  $n$  cũng có thể được **biểu diễn trên đồ thị  $Q_n$**  bằng cách đánh dấu các đỉnh tương ứng với các bộ  $(x_1, x_2, \dots, x_n)$  thỏa mãn  $F(x_1, \dots, x_n) = 1$

## 3 Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

# Hàm Boole

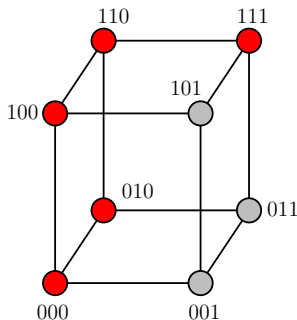


Đại số Boole  
Hoàng Anh Đức

## Ví dụ 1

Hàm  $F(x, y, z) = xy + \bar{z}$

| $x$ | $y$ | $z$ | $xy$ | $\bar{z}$ | $F(x, y, z)$ |
|-----|-----|-----|------|-----------|--------------|
| 1   | 1   | 1   | 1    | 0         | 1            |
| 1   | 1   | 0   | 1    | 1         | 1            |
| 1   | 0   | 1   | 0    | 0         | 0            |
| 1   | 0   | 0   | 0    | 1         | 1            |
| 0   | 1   | 1   | 0    | 0         | 0            |
| 0   | 1   | 0   | 0    | 1         | 1            |
| 0   | 0   | 1   | 0    | 0         | 0            |
| 0   | 0   | 0   | 0    | 1         | 1            |



## 4 Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

## Bài tập 3

Lập bảng giá trị của các hàm Boole sau:

(a)  $F(x, y, z) = x\bar{y}z + \bar{x}\bar{y}\bar{z}$

(b)  $F(x, y, z) = x(yz + \bar{y}\bar{z})$

## Bài tập 4

Biểu diễn mỗi hàm ở Bài tập 3 trên một khối lập phương  $Q_3$  bằng cách đánh dấu các đỉnh tại đó hàm đạt giá trị 1

# Hàm Boole



Đại số Boole  
Hoàng Anh Đức

## 5 Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

- Các hàm Boole bậc  $n$   $F$  và  $G$  được gọi là **bằng nhau (equal)** khi và chỉ khi  $F(b_1, b_2, \dots, b_n) = G(b_1, b_2, \dots, b_n)$  với mọi  $b_1, b_2, \dots, b_n \in B$
- Hai biểu thức Boole biểu diễn cùng một hàm Boole được gọi là hai biểu thức **tương đương (equivalent)**
- **Phần bù (complement)** của hàm Boole  $F$  là hàm  $\overline{F}$  được định nghĩa bởi  $\overline{F}(x_1, x_2, \dots, x_n) = \overline{F(x_1, x_2, \dots, x_n)}$
- **Tổng Boole (Boolean sum)** của hai hàm  $F$  và  $G$  là hàm  $F + G$  được định nghĩa bởi  $(F + G)(x_1, x_2, \dots, x_n) = F(x_1, x_2, \dots, x_n) + G(x_1, x_2, \dots, x_n)$
- **Tích Boole (Boolean product)** của hai hàm  $F$  và  $G$  là hàm  $F \cdot G$  được định nghĩa bởi  $(F \cdot G)(x_1, x_2, \dots, x_n) = F(x_1, x_2, \dots, x_n) \cdot G(x_1, x_2, \dots, x_n)$

## Bài tập 5

Có tất cả bao nhiêu hàm Boole bậc  $n$  khác nhau?

# Hàm Boole



Đại số Boole  
Hoàng Anh Đức

## Một số hằng đẳng thức quan trọng trong Đại số Boole

| Tên gọi                                            | Đẳng thức                                         |
|----------------------------------------------------|---------------------------------------------------|
| Luật phần bù kép<br>(Law of the double complement) | $\overline{\overline{x}} = x$                     |
| Luật lũy đẳng<br>(Idempotent laws)                 | $x + x = x$<br>$x \cdot x = x$                    |
| Luật đồng nhất<br>(Identity laws)                  | $x + 0 = x$<br>$x \cdot 1 = x$                    |
| Luật nuốt<br>(Domination laws)                     | $x + 1 = 1$<br>$x \cdot 0 = 0$                    |
| Luật giao hoán<br>(Commutative laws)               | $x + y = y + x$<br>$xy = yx$                      |
| Luật kết hợp<br>(Associative laws)                 | $(x + y) + z = x + (y + z)$<br>$x(yz) = (xy)z$    |
| Luật phân phối<br>(Distributive laws)              | $x + yz = (x + y)(x + z)$<br>$x(y + z) = xy + xz$ |

### 6 Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

# Hàm Boole



Đại số Boole  
Hoàng Anh Đức

Một số hằng đẳng thức quan trọng trong Đại số Boole (tiếp)

| Tên gọi                             | Đẳng thức                                                                       |
|-------------------------------------|---------------------------------------------------------------------------------|
| Luật De Morgan<br>(De Morgan laws)  | $\overline{(xy)} = \bar{x} + \bar{y}$<br>$\overline{(x + y)} = \bar{x} \bar{y}$ |
| Luật hấp thụ<br>(Absorption laws)   | $x + xy = x$<br>$x(x + y) = x$                                                  |
| Tính chất đơn vị<br>(unit property) | $x + \bar{x} = 1$                                                               |
| Tính chất không<br>(zero property)  | $x\bar{x} = 0$                                                                  |

## Bài tập 6

Bằng cách sử dụng bảng để biểu diễn các hàm Boole, hãy chứng minh luật phân phối  $x + yz = (x + y)(x + z)$

## Bài tập 7

Bằng cách sử dụng các đẳng thức quan trọng khác trong bảng, hãy chứng minh luật hấp thụ  $x(x + y) = x$

### 7 Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

# Hàm Boole



Đại số Boole  
Hoàng Anh Đức

## 8 Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

- **Đối ngẫu (dual)** của một biểu thức Boole là biểu thức thu được bằng cách thay tổng (tích) bằng tích (tổng) và thay 0 (1) bằng 1 (0)

$$\blacksquare + \Rightarrow \cdot$$

$$\blacksquare 0 \Rightarrow 1$$

$$\blacksquare \cdot \Rightarrow +$$

$$\blacksquare 1 \Rightarrow 0$$

- Đối ngẫu của một hàm Boole  $F$  được biểu diễn bởi một biểu thức Boole là một hàm Boole được biểu diễn bởi đối ngẫu của biểu thức đó
- Đối ngẫu của một hàm Boole  $F$  được ký hiệu là  $F^d$  và không phụ thuộc vào biểu thức Boole cụ thể biểu diễn hàm Boole đó
- **Nguyên lý đối ngẫu (duality principle):** đẳng thức vẫn đúng khi lấy đối ngẫu cả hai vế

## Ví dụ 2

- Đối ngẫu của  $x(y + 0)$  là  $x + (y \cdot 1)$
- Đối ngẫu của  $\bar{x} \cdot 1 + (\bar{y} + z)$  là  $(\bar{x} + 0) \cdot (\bar{y} \cdot z)$

## Ví dụ 3

- Ta có  $x(x + y) = x$  (luật hấp thụ)
- Lấy đối ngẫu cả hai vế, ta có  $x + xy = x$  (luật hấp thụ)

# Hàm Boole



Đại số Boole  
Hoàng Anh Đức

- Các định nghĩa và kết quả chúng ta đã đề cập có thể được chuyển sang các kết quả cho các mệnh đề hoặc tập hợp
- Do đó, sẽ rất hữu ích nếu ta có thể định nghĩa đại số Boole một cách trừu tượng

## Đại số Boole

Một **Đại số Boole** (*Boolean algebra*) là một tập  $B$  với hai toán tử hai ngôi (binary operation)  $\vee$  và  $\wedge$ , các phần tử  $0$  và  $1$ , và một toán tử một ngôi (unary operation)  $\neg$ , ký hiệu  $[B, \vee, \wedge, \neg, 0, 1]$ , sao cho các tính chất sau đúng với mọi  $x, y$  và  $z$  thuộc  $B$

- Luật đồng nhất (identity laws):  $x \vee 0 = x$  và  $x \wedge 1 = x$
- Luật bù (complement laws):  $x \vee \bar{x} = 1$  và  $x \wedge \bar{x} = 0$
- Luật kết hợp (associative laws):  $(x \vee y) \vee z = x \vee (y \vee z)$  và  $(x \wedge y) \wedge z = x \wedge (y \wedge z)$
- Luật giao hoán (commutative laws):  $x \vee y = y \vee x$  và  $x \wedge y = y \wedge x$
- Luật phân phối (distributive laws):  $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$  và  $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$

### 9 Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole



- $[B, +, \cdot, -, 0, 1]$ , trong đó  $B = \{0, 1\}$  và  $+, \cdot, -$  là các toán tử đã định nghĩa ở phần đầu bài giảng, là một đại số Boole
- $[P, \vee, \wedge, \neg, \mathbf{F}, \mathbf{T}]$ , trong đó  $P$  là tập các mệnh đề logic và  $\vee, \wedge, \neg$  là các toán tử logic, là một đại số Boole
- $[\mathcal{P}(U), \cup, \cap, \overline{\phantom{x}}, \emptyset, U]$ , trong đó  $U$  là tập vũ trụ nào đó,  $\mathcal{P}(U)$  là tập tất cả các tập con của  $U$ , và  $\cup, \cap, \overline{\phantom{x}}$  là các toán tử tập hợp, là một đại số Boole



## Ví dụ 4

Sử dụng định nghĩa, chứng minh rằng trong một đại số Boole  $[B, \vee, \wedge, \neg, 0, 1]$ , với mọi  $x \in B$  ta có  $x \vee x = x$  và  $x \wedge x = x$  (luật lũy đẳng (idempotent laws))

$$x \vee x = (x \vee x) \wedge 1$$

$$= (x \vee x) \wedge (x \vee \bar{x})$$

$$= x \vee (x \wedge \bar{x})$$

$$= x \vee 0$$

$$= x$$

luật đồng nhất

luật bù

luật phân phối

luật bù

luật đồng nhất

$$x \wedge x = (x \wedge x) \vee 0$$

$$= (x \wedge x) \vee (x \wedge \bar{x})$$

$$= x \wedge (x \vee \bar{x})$$

$$= x \wedge 1$$

$$= x$$

luật đồng nhất

luật bù

luật phân phối

luật bù

luật đồng nhất

### 11 Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

# Biểu diễn các hàm Boole



Đại số Boole

Hoàng Anh Đức

Hàm Boole

12 Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

Hai bài toán quan trọng trong đại số Boole

- Cho các giá trị của một hàm Boole, tìm một biểu thức Boole biểu diễn hàm đó
  - Mọi hàm Boole có thể được biểu diễn bằng biểu thức *chỉ có các toán tử  $\vee, \wedge$ , và  $\neg$*
- Biểu diễn mọi hàm Boole bằng biểu thức với càng ít toán tử càng tốt
  - Mọi hàm Boole có thể được biểu diễn bằng biểu thức *chỉ có một toán tử*

# Biểu diễn các hàm Boole



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

13 Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

- Một biến Boole hoặc phần bù của nó được gọi là một **tục biến (literal)**
- Tích Boole  $y_1 y_2 \dots y_n$  trong đó  $y_i = x_i$  hoặc  $y_i = \overline{x_i}$  với các biến Boole  $x_1, x_2, \dots, x_n$  được gọi là một **tiểu hạng (minterm)**. Nói cách khác, mỗi tiểu hạng là tích của các tục biến
- Tiểu hạng  $y_1 y_2 \dots y_n$  có giá trị bằng 1 khi và chỉ khi  $y_i = 1$  với mọi  $1 \leq i \leq n$ , nghĩa là khi và chỉ khi  $y_i = x_i$  nếu  $x_i = 1$  và  $y_i = \overline{x_i}$  nếu  $x_i = 0$
- Tổng Boole của các tiểu hạng có giá trị bằng 1 chỉ khi một trong các tiểu hạng của tổng có giá trị 1

# Biểu diễn các hàm Boole

## Ví dụ 5

Tìm biểu thức Boole biểu diễn các hàm  $F$  và  $G$  có các giá trị cho bởi bảng sau

- Một biểu thức biểu diễn  $F$  cần có giá trị 1 khi  $x = z = 1$  và  $y = 0$ , và có giá trị 0 trong các trường hợp còn lại

- Lấy  $F(x, y, z) = x\bar{y}z$

- Một biểu thức biểu diễn  $G$  cần có giá trị 1 khi  $x = y = 1$  và  $z = 0$  hoặc  $x = z = 0$  và  $y = 1$ , và có giá trị 0 trong các trường hợp còn lại

- Lấy  $G(x, y, z) = xy\bar{z} + \bar{x}y\bar{z}$

Các biểu thức như trên được gọi là *khai triển tổng các tích (sum-of-products expansion)* hay *dạng chuẩn tắc (Disjunctive Normal Form - DNF)* của các hàm  $F$  và  $G$

| $x$ | $y$ | $z$ | $F$ | $G$ |
|-----|-----|-----|-----|-----|
| 1   | 1   | 1   | 0   | 0   |
| 1   | 1   | 0   | 0   | 1   |
| 1   | 0   | 1   | 1   | 0   |
| 1   | 0   | 0   | 0   | 0   |
| 0   | 1   | 1   | 0   | 0   |
| 0   | 1   | 0   | 0   | 1   |
| 0   | 0   | 1   | 0   | 0   |
| 0   | 0   | 0   | 0   | 0   |

Đại số Boole

Hoàng Anh Đức

Hàm Boole

14 Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

# Biểu diễn các hàm Boole



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

15 Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

Một *khai triển tổng các tích (sum-of-products expansion)* hay *dạng tuyển chuẩn tắc (Disjunctive Normal Form - DNF)* của hàm Boole  $F$  là một biểu diễn của  $F$  dưới dạng tổng các tiểu hạng. Ngoài cách dùng bảng các giá trị, ta cũng có thể tìm dạng khai triển tổng các tích thông qua các đẳng thức đã biết

## Ví dụ 6

Tìm khai triển tổng các tích của  $F(x, y, z) = (x + y)\bar{z}$

$$(x + y)\bar{z} = x\bar{z} + y\bar{z}$$

$$= x1\bar{z} + 1y\bar{z}$$

$$= x(y + \bar{y})\bar{z} + (x + \bar{x})y\bar{z}$$

$$= xy\bar{z} + x\bar{y}\bar{z} + xy\bar{z} + \bar{x}y\bar{z}$$

$$= xy\bar{z} + x\bar{y}\bar{z} + \bar{x}y\bar{z}$$

luật phân phối

luật đồng nhất

tính chất đơn vị

luật phân phối

luật lũy đẳng

# Biểu diễn các hàm Boole

Tương tự, ta cũng có thể tìm các biểu thức biểu diễn một hàm Boole  $F$  bằng cách lấy tích Boole của các tổng Boole. Biểu thức tìm được gọi là *khai triển tích các tổng (product-of-sums expansion)* hay *dạng hội chuẩn tắc (conjunctive normal form - CNF)* của  $F$

## Ví dụ 7

Tìm biểu thức Boole biểu diễn các hàm  $F$  và  $G$  có các giá trị cho bởi bảng sau

- Một biểu thức biểu diễn  $F$  cần có giá trị 0 khi  $x = z = 1$  và  $y = 0$ , và có giá trị 1 trong các trường hợp còn lại
- Lấy  $F(x, y, z) = \bar{x} + y + \bar{z}$
- Một biểu thức biểu diễn  $G$  cần có giá trị 0 khi  $x = y = 1$  và  $z = 0$  hoặc  $x = z = 0$  và  $y = 1$ , và có giá trị 1 trong các trường hợp còn lại
- Lấy  $G(x, y, z) = (\bar{x} + \bar{y} + z)(x + \bar{y} + z)$

| $x$ | $y$ | $z$ | $F$ | $G$ |
|-----|-----|-----|-----|-----|
| 1   | 1   | 1   | 1   | 1   |
| 1   | 1   | 0   | 1   | 0   |
| 1   | 0   | 1   | 0   | 1   |
| 1   | 0   | 0   | 1   | 1   |
| 0   | 1   | 1   | 1   | 1   |
| 0   | 1   | 0   | 1   | 0   |
| 0   | 0   | 1   | 1   | 1   |
| 0   | 0   | 0   | 1   | 1   |

Đại số Boole

Hoàng Anh Đức

Hàm Boole

16 Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

# Biểu diễn các hàm Boole



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

17 Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

## Bài tập 8

Tìm một biểu diễn của hàm Boole  $F(x, y, z)$  biết rằng  $F$  nhận giá trị 1 khi và chỉ khi:

- (a) có đúng 2 trong 3 biến  $x, y, z$  nhận giá trị 1.
- (b) có một số chẵn biến nhận giá trị 1.

## Bài tập 9

Tìm biểu diễn dưới dạng tổng của các tiểu hạng (minterm) của các hàm sau:

- (a)  $F(x, y, z) = (x + \bar{z})y$
- (b)  $F(x, y, z) = x\bar{y} + y\bar{z}$

# Biểu diễn các hàm Boole



Đại số Boole

Hoàng Anh Đức

Hàm Boole

18 Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

- Mỗi hàm Boole đều có thể được biểu diễn dưới dạng tổng Boole của các tiểu hạng, trong đó mỗi tiểu hạng là một tích Boole của các biến Boole hoặc các phần bù của chúng
- Nói cách khác, mỗi hàm Boole có thể được biểu diễn bằng cách dùng các toán tử Boole  $+$ ,  $\cdot$ ,  $-$ . Tập các toán tử  $\{+, \cdot, -\}$  được gọi là một tập **đầy đủ (functionally complete)**
- Liệu ta có thể tìm được một tập các toán tử đầy đủ nhỏ hơn hay không?
  - Tập  $\{+, -\}$  là một tập đầy đủ (luật De Morgan)
  - Tập  $\{\cdot, -\}$  là một tập đầy đủ (luật De Morgan)
  - Tập  $\{+, \cdot\}$  không là một tập đầy đủ (ví dụ, ta không thể biểu diễn  $F(x) = \bar{x}$  bằng một biểu thức Boole chỉ có hai phép toán  $+$  và  $\cdot$ )
  - Liệu có tồn tại một tập các toán tử đầy đủ **chỉ có đúng một phần tử** hay không?

# Biểu diễn các hàm Boole



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

19 Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

Ta định nghĩa hai toán tử  $|$  hay NAND và  $\downarrow$  hay NOR

| $x$ | $y$ | $x   y$ ( $x$ NAND $y$ ) | $x \downarrow y$ ( $x$ NOR $y$ ) |
|-----|-----|--------------------------|----------------------------------|
| 1   | 1   | 0                        | 0                                |
| 1   | 0   | 1                        | 0                                |
| 0   | 1   | 1                        | 0                                |
| 0   | 0   | 1                        | 1                                |

Các tập  $\{| \}$  và  $\{ \downarrow \}$  đều là các tập đầy đủ

## Bài tập 10

Chứng minh rằng

(a)  $\bar{x} = x | x$

(b)  $xy = (x | y) | (x | y)$

(c)  $x + y = (x | x) | (y | y)$

## Bài tập 11

Chứng minh rằng

(a)  $\bar{x} = x \downarrow x$

(b)  $xy = (x \downarrow x) \downarrow (y \downarrow y)$

(c)  $x + y = (x \downarrow y) \downarrow (x \downarrow y)$

# Biểu diễn các hàm Boole



Đại số Boole

Hoàng Anh Đức

Hàm Boole

20 Biểu diễn các hàm Boole

Các cổng logic

Rút gọn hàm Boole

## Bài tập 12

Sử dụng luật De Morgan, ta có thể biểu diễn phép toán  $+$  theo hai phép toán  $\cdot$  và  $\bar{\phantom{x}}$ , từ đó có thể biểu diễn mọi hàm Boole bằng một biểu thức chỉ chứa hai phép toán đó. Hãy tìm một biểu diễn như vậy của các hàm sau:

(a)  $F(x, y) = x + y$

(b)  $F(x, y, z) = \bar{x}(x + \bar{y}) + z$

## Bài tập 13

Tương tự bài trước, ta cũng có thể biểu diễn mọi hàm Boole bằng một biểu thức chỉ chứa hai phép toán  $+$  và  $\bar{\phantom{x}}$ . Tìm một biểu diễn như vậy của các hàm sau:

(a)  $F(x, y) = \overline{xy}$

(b)  $F(x, y, z) = x + \bar{y}(\bar{x} + z)$

# Các cổng logic



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

21 Các cổng logic

Rút gọn hàm Boole

- Đại số Boole có thể được sử dụng trong việc mô hình hóa sơ đồ các mạch trong các dụng cụ điện tử
- Mỗi một đầu vào và một đầu ra của một dụng cụ như vậy có thể được xem như một phần tử của tập  $\{0, 1\}$
- Một dụng cụ được tạo bởi các **mạch (circuit)** khác nhau. Các phần tử cơ bản của các mạch này được gọi là các **cổng (gate)**. Mỗi loại cổng thực hiện một toán tử Boole
- Trong bài giảng, chúng ta sẽ
  - Định nghĩa một số loại cổng
  - Sử dụng các cổng này và các quy tắc trong Đại số Boole để thiết kế các mạch thực hiện nhiệm vụ nào đó
  - Các mạch mà ta sẽ đề cập sẽ cho ra đầu ra chỉ phụ thuộc vào đầu vào chứ không phụ thuộc vào trạng thái hiện thời của mạch. Nói cách khác, các mạch này không có khả năng nhớ. Những mạch như vậy gọi là các **mạch tổ hợp (combinatorial circuit)**

# Các cổng logic



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

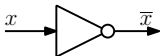
Biểu diễn các hàm Boole

22 Các cổng logic

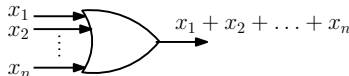
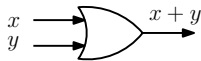
Rút gọn hàm Boole

- **Bộ đảo (inverter):** đầu vào là giá trị của một biến Boole và đầu ra là phần bù của giá trị đầu vào
- **Cổng OR (OR gate):** đầu vào là giá trị của hai hay nhiều biến Boole và đầu ra là tổng Boole của các giá trị đó
- **Cổng AND (AND gate):** đầu vào là giá trị của hai hay nhiều biến Boole và đầu ra là tích Boole của các giá trị đó

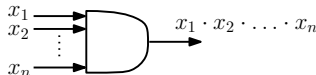
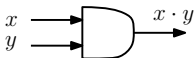
Bộ đảo



Cổng OR



Cổng AND



# Các cổng logic



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm  
Boole

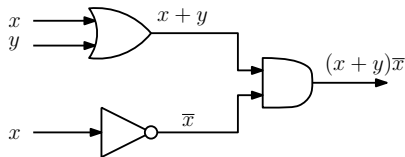
23 Các cổng logic

Rút gọn hàm Boole

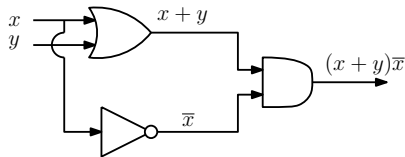
## Ví dụ 8

Mạch tổ hợp có đầu ra là  $(x + y)\bar{x}$

Cách 1



Cách 2



## Bài tập 14

Vẽ các mạch tổ hợp có đầu ra

(a)  $\bar{x}(y + \bar{z})$

(b)  $(x + y + z)(\bar{x} + \bar{y} + \bar{z})$



## Ví dụ 9

Đôi khi một hệ thống đèn cố định được điều khiển bởi hai hay nhiều công tắc. Các mạch điện cần được thiết kế sao cho khi ấn một công tắc bất kỳ thì hệ thống đèn đang bật sẽ tắt và hệ thống đèn đang tắt sẽ bật. Ta thiết kế một mạch thực hiện điều đó khi có hai công tắc

- Giả sử  $x = 1$  khi công tắc thứ nhất đóng và  $x = 0$  khi nó mở. Tương tự,  $y = 1$  khi công tắc thứ hai đóng và  $y = 0$  khi nó mở
- Giả sử  $F(x, y) = 1$  khi đèn sáng và  $F(x, y) = 0$  khi đèn tắt
- Ta chọn để đèn sáng khi cả hai công tắc đóng, nghĩa là  $F(x, y) = 1$  khi  $x = y = 1$ . Theo yêu cầu ở trên, khi một trong hai công tắc mở, đèn sẽ tắt, nghĩa là  $F(x, y) = 0$  khi  $x = 1$  và  $y = 0$  hoặc  $x = 0$  và  $y = 1$ . Cuối cùng, khi cả hai công tắc mở, đèn sẽ lại sáng, nghĩa là  $F(x, y) = 1$  khi  $x = y = 0$
- Một biểu diễn của  $F(x, y)$  là  $xy + \bar{x}\bar{y}$

# Các cổng logic



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

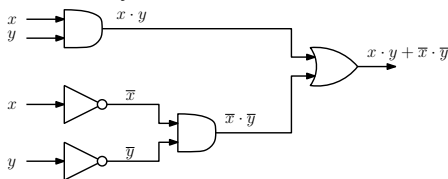
Biểu diễn các hàm Boole

25 Các cổng logic

Rút gọn hàm Boole

Mạch với hai công tắc thiết kế theo yêu cầu là

| $x$ | $y$ | $F(x, y)$ |
|-----|-----|-----------|
| 1   | 1   | 1         |
| 1   | 0   | 0         |
| 0   | 1   | 0         |
| 0   | 0   | 1         |



## Bài tập 15

Hãy thử thiết kế mạch theo yêu cầu trong Ví dụ 9 với ba công tắc

# Rút gọn hàm Boole



Đại số Boole

Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

26

Rút gọn hàm Boole

- Hiệu quả của một mạch tổ hợp phụ thuộc vào số các cổng và sự bố trí các cổng đó
- Chúng ta luôn có thể sử dụng khai triển tổng các tích của mạch để tìm tập các cổng logic thực hiện mạch đó. Tuy nhiên, khai triển tổng các tích có thể chứa nhiều số hạng hơn mức cần thiết
- Việc **rút gọn (minimization)** hàm Boole nghĩa là tìm dạng biểu thức Boole đơn giản nhất để biểu diễn hàm Boole đó

## Ví dụ 10

- Xét mạch có đầu ra bằng 1 khi và chỉ khi  $x = y = z = 1$  hoặc  $x = z = 1$  và  $y = 0$
- Khai triển tổng các tích của mạch này là  $xyz + x\bar{y}z$ . Để biểu diễn mạch này, ta cần ba cổng và một bộ đảo
- Mặt khác, ta có thể viết  $xyz + x\bar{y}z = xz(y + \bar{y}) = xz$ . Lúc này, ta có thể biểu diễn mạch chỉ với một cổng

37

# Rút gọn hàm Boole



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

27 Rút gọn hàm Boole

Một số phương pháp rút gọn hàm Boole

- *Phương pháp biến đổi đại số* dựa vào các đẳng thức đã biết
- *Phương pháp sơ đồ Karnaugh (Karnaugh Maps)*
- *Phương pháp Quine-McCluskey (Quine-McCluskey Method)*

Nguyên lý cơ bản của các phương pháp rút gọn trên được thể hiện trong đẳng thức

$$x \cdot y + x \cdot \bar{y} = x \cdot (y + \bar{y}) = x$$

# Rút gọn hàm Boole



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

28 Rút gọn hàm Boole

- **Phương pháp sơ đồ Karnaugh (Karnaugh Maps)** thường chỉ được áp dụng cho các hàm Boole có 6 biến hoặc ít hơn
- **Ý tưởng:** là tìm các số hạng trong biểu diễn của hàm Boole có thể tổ hợp được

Sơ đồ Karnaugh cho hàm Boole có 2 biến

- Bảng gồm 4 ô vuông như trong hình bên. Nếu các tiểu hạng có mặt trong biểu thức biểu diễn hàm Boole thì viết 1 vào các ô tương ứng
- Hai ô kề nhau nếu các tiểu hạng tương ứng chỉ khác nhau một tục biến
- Bất cứ khi nào có 2 ô kề nhau có số 1 thì các tiểu hạng được biểu diễn bởi các ô đó có thể được tổ hợp lại thành một tục biến
- Nếu cả 4 ô đều có số 1 thì các tiểu hạng được biểu diễn bởi các ô đó có thể được tổ hợp lại thành một số hạng duy nhất: biểu diễn Boole của 1

|           | $y$               | $\bar{y}$               |
|-----------|-------------------|-------------------------|
| $x$       | $x \cdot y$       | $x \cdot \bar{y}$       |
| $\bar{x}$ | $\bar{x} \cdot y$ | $\bar{x} \cdot \bar{y}$ |

# Rút gọn hàm Boole



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

29 Rút gọn hàm Boole

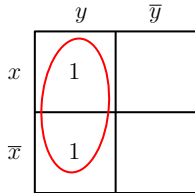
## Ví dụ 11

Rút gọn các khai triển tổng các tích sau bằng sơ đồ Karnaugh

(a)  $xy + \bar{x}y$

(b)  $x\bar{y} + \bar{x}y$

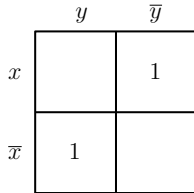
(c)  $x\bar{y} + \bar{x}y + \bar{x}\bar{y}$



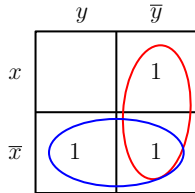
(a)  $xy + \bar{x}y$



$y$



(b)  $x\bar{y} + \bar{x}y$



(c)  $x\bar{y} + \bar{x}y + \bar{x}\bar{y}$



$\bar{x} + \bar{y}$

# Rút gọn hàm Boole



Đại số Boole

Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

30 Rút gọn hàm Boole

Sơ đồ Karnaugh cho hàm Boole có 3 biến

- Bảng gồm 8 ô vuông như trong hình dưới. Nếu các tiểu hạng có mặt trong biểu thức biểu diễn hàm Boole thì viết 1 vào các ô tương ứng
- Hai ô kề nhau nếu các tiểu hạng tương ứng chỉ khác nhau một tực biến
- Các tiểu hạng của 2 ô kề nhau có số 1 có thể tổ hợp thành tích của hai tực biến
- Các tiểu hạng của 4 ô kề nhau có số 1 có thể tổ hợp thành một tực biến duy nhất
- Các tiểu hạng của 8 ô kề nhau có số 1 tổ hợp lại thành một số hạng duy nhất: biểu diễn Boole của 1

|           | $yz$        | $y\bar{z}$        | $\bar{y}\bar{z}$        | $\bar{y}z$        |
|-----------|-------------|-------------------|-------------------------|-------------------|
| $x$       | $xyz$       | $xy\bar{z}$       | $x\bar{y}\bar{z}$       | $x\bar{y}z$       |
| $\bar{x}$ | $\bar{x}yz$ | $\bar{x}y\bar{z}$ | $\bar{x}\bar{y}\bar{z}$ | $\bar{x}\bar{y}z$ |

# Rút gọn hàm Boole



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

31 Rút gọn hàm Boole

## Ví dụ 12

|           | $yz$ | $y\bar{z}$ | $\bar{y}z$ | $\bar{y}\bar{z}$ |
|-----------|------|------------|------------|------------------|
| $x$       |      |            |            |                  |
| $\bar{x}$ |      |            |            |                  |



$\bar{y}z$

|           | $yz$ | $y\bar{z}$ | $\bar{y}z$ | $\bar{y}\bar{z}$ |
|-----------|------|------------|------------|------------------|
| $x$       |      |            |            |                  |
| $\bar{x}$ |      |            |            |                  |



$\bar{z}$

|           | $yz$ | $y\bar{z}$ | $\bar{y}z$ | $\bar{y}\bar{z}$ |
|-----------|------|------------|------------|------------------|
| $x$       |      |            |            |                  |
| $\bar{x}$ |      |            |            |                  |



$\bar{x}z$

|           | $yz$ | $y\bar{z}$ | $\bar{y}z$ | $\bar{y}\bar{z}$ |
|-----------|------|------------|------------|------------------|
| $x$       |      |            |            |                  |
| $\bar{x}$ |      |            |            |                  |



$\bar{x}$

# Rút gọn hàm Boole



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

32 Rút gọn hàm Boole

- Phương pháp sơ đồ Karnaugh rất khó sử dụng khi số biến của hàm Boole lớn hơn 4. Thêm vào đó, phương pháp này cần rà soát trực quan để nhận dạng các số hạng cần nhóm lại
- **Phương pháp Quine-McCluskey (Quine-McCluskey Method)** khắc phục được những nhược điểm trên
  - Tìm các số hạng là các ứng viên để đưa vào khai triển rút gọn như một tổng các tích Boole
  - Xác định xem trong số các ứng viên đó, các số hạng nào là thực sự dùng được

# Rút gọn hàm Boole



Đại số Boole

Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

33

Rút gọn hàm Boole

## Ví dụ 13

Rút gọn  $xyz + x\bar{y}z + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$  bằng phương pháp Quine-McCluskey

- Biểu diễn mỗi tiểu hạng bằng một chuỗi nhị phân: thay mỗi biến bằng 1 và mỗi phủ định của biến bằng 0

| Tiểu hạng               | Chuỗi nhị phân | Số các số 1 |
|-------------------------|----------------|-------------|
| $xyz$                   | 111            | 3           |
| $x\bar{y}z$             | 101            | 2           |
| $\bar{x}yz$             | 011            | 2           |
| $\bar{x}\bar{y}z$       | 001            | 1           |
| $\bar{x}\bar{y}\bar{z}$ | 000            | 0           |

- Nhóm các chuỗi nhị phân theo số các số 1

# Rút gọn hàm Boole



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

34 Rút gọn hàm Boole

- Tổ hợp các số hạng có thể và xây dựng các bảng dưới đây theo các nguyên tắc sau
  - Các tiểu hạng có thể tổ hợp lại nếu chúng chỉ khác nhau một tục biến. Do đó, hai số hạng có thể tổ hợp được thì *các chuỗi nhị phân tương ứng với chúng chỉ khác nhau một số 1*.
  - Dấu “-” biểu thị một biến không xuất hiện trong kết quả thu được sau khi tổ hợp. Dấu “✓” biểu thị số hạng có thể tổ hợp với số hạng khác

| Số hạng |                         | Chuỗi bit |   |
|---------|-------------------------|-----------|---|
| 1       | $xyz$                   | 111       | ✓ |
| 2       | $x\bar{y}z$             | 101       | ✓ |
| 3       | $\bar{x}yz$             | 011       | ✓ |
| 4       | $\bar{x}\bar{y}z$       | 001       | ✓ |
| 5       | $\bar{x}\bar{y}\bar{z}$ | 000       | ✓ |



| Số hạng |                  | Chuỗi bit |   |
|---------|------------------|-----------|---|
| (1, 2)  | $xz$             | 1-1       | ✓ |
| (1, 3)  | $yz$             | -11       | ✓ |
| (2, 4)  | $\bar{y}z$       | -01       | ✓ |
| (3, 4)  | $\bar{x}z$       | 0-1       | ✓ |
| (4, 5)  | $\bar{x}\bar{y}$ | 00-       |   |



| Số hạng      |     | Chuỗi bit |
|--------------|-----|-----------|
| (1, 2, 3, 4) | $z$ | --1       |
| (1, 3, 2, 4) | $z$ | --1       |

# Rút gọn hàm Boole



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

35 Rút gọn hàm Boole

- Trong các bảng trên, chúng ta đã chỉ ra các số hạng được sử dụng để tạo ra các tích có số tục biến nhỏ hơn nhưng không nhất thiết có mặt trong biểu diễn rút gọn hàm Boole
- Ta xây dựng biểu thức rút gọn của hàm Boole theo nguyên tắc sau:
  - Xét các tích chưa được sử dụng để xây dựng các tích có số tục biến nhỏ hơn (các số hạng không có đánh dấu ✓ tương ứng):  $z$  và  $\bar{x}\bar{y}$
  - Lập bảng kiểm tra xem các ứng viên  $z$  và  $\bar{x}\bar{y}$  có thực sự phủ hết các tiểu hạng gốc của hàm  $F(x, y, z)$  ban đầu hay không

|                  | $xyz$ | $x\bar{y}z$ | $\bar{x}yz$ | $\bar{x}\bar{y}z$ | $\bar{x}\bar{y}\bar{z}$ |
|------------------|-------|-------------|-------------|-------------------|-------------------------|
| $z$              | X     | X           | X           | X                 |                         |
| $\bar{x}\bar{y}$ |       |             |             | X                 | X                       |

- Do đó, kết quả cuối cùng là  $F(x, y, z) = z + \bar{x}\bar{y}$

# Rút gọn hàm Boole



Đại số Boole

Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm Boole

Các cổng logic

36 Rút gọn hàm Boole

## Ví dụ 14

Sử dụng phương pháp Quine-McCluskey để rút gọn biểu diễn của  $F(w, x, y, z)$  sau  $wxy\bar{z} + w\bar{x}yz + w\bar{x}y\bar{z} + \bar{w}xyz + \bar{w}x\bar{y}z + \bar{w}\bar{x}yz + \bar{w}\bar{x}\bar{y}z$

| Số hạng | Chuỗi bit                |        | Số hạng | Chuỗi bit                |
|---------|--------------------------|--------|---------|--------------------------|
| 1       | $wxy\bar{z}$             | 1110 ✓ | (1, 4)  | $wy\bar{z}$ 1-10         |
| 2       | $w\bar{x}yz$             | 1011 ✓ | (2, 4)  | $w\bar{x}y$ 101-         |
| 3       | $\bar{w}xyz$             | 0111 ✓ | (2, 6)  | $\bar{x}yz$ -011         |
| 4       | $w\bar{x}y\bar{z}$       | 1010 ✓ | (3, 5)  | $\bar{w}xz$ 01-1 ✓       |
| 5       | $\bar{w}\bar{x}yz$       | 0101 ✓ | (3, 6)  | $\bar{w}yz$ 0-11 ✓       |
| 6       | $\bar{w}\bar{x}y\bar{z}$ | 0011 ✓ | (5, 7)  | $\bar{w}\bar{y}z$ 0-01 ✓ |
| 7       | $\bar{w}\bar{x}\bar{y}z$ | 0001 ✓ | (6, 7)  | $\bar{w}\bar{x}z$ 00-1 ✓ |



| Số hạng      | Chuỗi bit        |
|--------------|------------------|
| (3, 5, 6, 7) | $\bar{w}z$ 0- -1 |
| (3, 6, 5, 7) | $\bar{w}z$ 0- -1 |

|             | $wxy\bar{z}$ | $w\bar{x}yz$ | $\bar{w}xyz$ | $w\bar{x}y\bar{z}$ | $\bar{w}\bar{x}yz$ | $\bar{w}\bar{x}\bar{y}z$ |
|-------------|--------------|--------------|--------------|--------------------|--------------------|--------------------------|
| $\bar{w}z$  |              |              | X            | X                  | X                  | X                        |
| $wy\bar{z}$ | X            |              |              | X                  |                    |                          |
| $w\bar{x}y$ |              | X            |              | X                  |                    |                          |
| $\bar{x}yz$ |              | X            |              |                    | X                  |                          |

Do đó,  $F(w, x, y, z) = \bar{w}z + wy\bar{z} + w\bar{x}y$  hoặc  $F(w, x, y, z) = \bar{w}z + wy\bar{z} + \bar{x}yz$

# Rút gọn hàm Boole



Đại số Boole  
Hoàng Anh Đức

Hàm Boole

Biểu diễn các hàm  
Boole

Các cổng logic

37 Rút gọn hàm Boole

## Bài tập 16

Dùng sơ đồ Karnaugh và phương pháp Quine-McCluskey để rút gọn các khai triển tổng các tích sau

(a)  $xy\bar{z} + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}\bar{z}$

(b)  $x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$

(c)  $xyz + xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}$

Part I

Phụ lục

# Nội dung



Đại số Boole

Hoàng Anh Đức

Một số lỗi thường gặp

Một số lỗi thường gặp

# Một số lỗi thường gặp



Đại số Boole  
Hoàng Anh Đức

## Chú ý

Tham khảo từ tài liệu “Common Mistakes in Discrete Mathematics” ([https://highereducation.com/sites/dl/free/125967651x/1106131/Common\\_Mistakes\\_in\\_Discrete\\_Math.pdf](https://highereducation.com/sites/dl/free/125967651x/1106131/Common_Mistakes_in_Discrete_Math.pdf))

2

Một số lỗi thường gặp

### (a) Sai lệch một cấp độ trừu tượng khi suy nghĩ về hàm Boolean

- Một hàm Boolean với  $n$  biến có thể được biểu diễn bằng một bảng với  $2^n$  hàng; do đó có  $2^{2^n}$  hàm Boolean khác nhau với  $n$  biến

### (b) Đặt bộ đảo (inverter) sai vị trí khi xây dựng mạch tổ hợp

- Nếu chúng ta muốn đảo giá trị đầu ra của một cổng, bộ đảo cần phải được đặt sau cổng đó

4

# Một số lỗi thường gặp (tiếp)



Đại số Boole  
Hoàng Anh Đức

3 Một số lỗi thường gặp

## (c) Quên áp dụng đúng luật De Morgan khi đánh giá đầu ra của một mạch tổ hợp

- Đầu ra của một mạch là một biểu thức Boolean nhất định của các biến đầu vào
- Khi đơn giản hóa biểu thức này, điều quan trọng là phải nhớ rằng  $\overline{x \cdot y} = \overline{x} + \overline{y}$  và  $\overline{x + y} = \overline{x} \cdot \overline{y}$

## (d) Không tìm thấy các khối lớn nhất có thể khi tìm kiếm biểu thức Boolean tối thiểu bằng sơ đồ Karnaugh

- Vì không có thuật toán hiệu quả nào được biết để giải quyết vấn đề này trong trường hợp tổng quát (với nhiều hơn một vài biến), nên không có gì đáng ngạc nhiên khi quy trình này dường như liên quan đến một số “đoán mò” không mấy đẹp đẽ

## (e) Không tìm thấy cách phủ tốt nhất khi tìm kiếm biểu thức Boolean tối thiểu bằng phương pháp Quine-McCluskey

# Một số lỗi thường gặp (tiếp)



Đại số Boole  
Hoàng Anh Đức

4

Một số lỗi thường gặp

- Vì không có thuật toán hiệu quả nào được biết để giải quyết vấn đề này trong trường hợp tổng quát (với nhiều hơn một vài biến), nên không có gì đáng ngạc nhiên khi quy trình này dường như liên quan đến một số “đoán mò” không mấy đẹp đẽ
- Có thể rất khó để đảm bảo rằng phương án phủ mà chúng ta tìm được với, chẳng hạn, năm minterm thực sự là tốt nhất có thể – rằng không có một phương án phủ nào khác chỉ với bốn minterm

4