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 24 tháng 5 năm 2024

Nội dung

1. Lôgic và Chứng minh
2. Các cấu trúc cơ bản: Tập hợp, Hàm, Dãy, Tổng
3. Quy nạp và Đệ quy
4. Thuật toán I: Giới thiệu, một số thuật toán tìm kiếm và sắp xếp, độ tăng của hàm
5. Thuật toán II: Độ phức tạp tính toán, 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

Lôgic mệnh đề
Mệnh đề

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
- ✔ $1 = 2$
- ✔ $(20 + 24) + (20 + 24)(20 + 24) + (20 + 24) = 2024$
- ✔ Mọi số chẵn lớn hơn hoặc bằng 4 là tổng của hai số nguyên tố (Giả thuyết Goldbach)
- ✖ Mấy giờ rồi? [Câu hỏi]
- ✖ Hãy đọc quyển sách này [Mệnh lệnh]
- ✖ Thời tiết hôm nay lạnh quá [Ý kiến, Cảm thán]
- ✖ $x + 1 = 2$ [Đúng/sai tùy vào x]

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* $x^2 + y^2 = z^2$

- Ta thường sử dụng các chữ cái $p, q, r, s, \ldots$ 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	$\leftrightarrow$

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

Ví dụ 1

Xét các mệnh đề sau:

(1) Hà Nội là thủ đô của Việt Nam [Mệnh đề nguyên tử]

(2) Hà Nội là thủ đô của Việt Nam và Hà Nội là một trong hai đô thị loại đặc biệt của Việt Nam [Mệnh đề phức hợp]

(3) Hà Nội là thủ đô và đồng thời là một trong hai đô thị loại đặc biệt của Việt Nam [Mệnh đề phức hợp]

- *Phủ định (negation)* của mệnh đề p, ký hiệu $\neg p$ hoặc $\overline{p}$, 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

- *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

- *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

- *Tuyển 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

- Mệnh đề *kéo theo (implication)* $p \to q$, với p, q là hai mệnh đề cho trước, là mệnh đề "nếu p, thì q". Giá trị chân lý $p \to q =$ F khi và chỉ khi $p =$ T và $q =$ F, và trong mọi trường hợp còn lại $p \to q =$ 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 $p \to q :=$ "Nếu 2 là số chẵn, thì 2 là số nguyên tố"
- Bảng chân trị

p	q	$p \to q$
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, $p \to q =$ F không nhất thiết là "từ p không suy ra q" mà đơn giản chỉ là $p =$ T và $q =$ F

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

Bài tập 2

Xây dựng bảng chân trị cho các mệnh đề trên. Có nhận xét gì về các giá trị của các mệnh đề này?

p	q	$\neg p$	$\neg q$	$p \to q$	$q \to p$	$\neg q \to \neg p$	$\neg p \to \neg q$
T	T			T			
T	F			F			
F	T			T			
F	F			T			

- 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

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

Ví dụ 2

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 3

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 mệnh đề
Lôgic và các toán tử bit

- Một *bit* (**bin**ary dig**it** = 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*: $\overline{}$ (NOT), $\wedge$ (AND), $\vee$ (OR), $\oplus$ (XOR)

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

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

- $\overline{x_1 \dots x_n} = (\overline{x_1}) \dots (\overline{x_n})$
- $x_1 \dots x_n \wedge y_1 \dots y_n = (x_1 \wedge y_1) \dots (x_n \wedge y_n)$
- $x_1 \dots x_n \vee y_1 \dots y_n = (x_1 \vee y_1) \dots (x_n \vee y_n)$
- $x_1 \dots x_n \oplus y_1 \dots y_n = (x_1 \oplus y_1) \dots (x_n \oplus y_n)$

Bài tập 3

(a) $\overline{11010} =$

(b) $11010 \vee 10001 =$

(c) $11010 \wedge 10001 =$

(d) $11010 \oplus 10001 =$

Các mệnh đề tương đương

Phân loại mệnh đề

- 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 $\mathbf{T}$
  - $p \vee \neg p$
- 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 $\mathbf{F}$
  - $p \wedge \neg p$
- Một *tiếp liên (contingency)* là một mệnh đề phức hợp không phải là hằng đúng cũng không phải là mâu thuẫn
  - $(p \vee q) \rightarrow r$

Bài tập 4

Xây dựng bảng chân trị cho các mệnh đề ví dụ trên

Các mệnh đề tương đương

Tương đương lôgic

- Mệnh đề phức hợp p *tương đương lôgic (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 lôgic 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

Ví dụ 4

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 5

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

Các mệnh đề tương đương
Tương đương lôgic

Một số tương đương lôgic quan trọng

Tên gọi	Tương đương lôgic
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)	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$

Một số tương đương lôgic quan trọng (tiếp)

Tên gọi	Tương đương lôgic
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 lôgic quan trọng ở trên, $\mathbf{T}$ là một mệnh đề phức hợp luôn đúng (hằng đúng) và $\mathbf{F}$ là một mệnh đề phức hợp luôn sai (mâu thuẫn)

Bài tập 6

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

Ví dụ 5

Chứng minh $\neg(p \lor (\neg p \land q))$ và $\neg p \land \neg q$ là tương đương lôgic bằng cách sử dụng các tương đương lôgic đã biết

$$
\begin{aligned}
\neg(p \lor (\neg p \land q)) &\equiv \neg((p \lor \neg p) \land (p \lor q)) && \text{Luật phân phối}\\
&\equiv \neg(\mathbf{T} \land (p \lor q)) && \text{Luật phủ định}\\
&\equiv \neg((p \lor q) \land \mathbf{T}) && \text{Luật giao hoán}\\
&\equiv \neg(p \lor q) && \text{Luật đồng nhất}\\
&\equiv \neg p \land \neg q && \text{Luật De Morgan}
\end{aligned}
$$

Bài tập 7

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

Các mệnh đề tương đương

Tương đương lôgic

- Một số tương đương lôgic liên quan đến phép kéo theo

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

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

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

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

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

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

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

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

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

- Một số tương đương lôgic liên quan đến phép tương đương
  (1) $p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$
  (2) $p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
  (3) $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
  (4) $\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Bài tập 8

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

Các mệnh đề tương đương

Tương đương lôgic

Ví dụ 6

Chứng minh $(p \land q) \to (p \lor q)$ là một hằng đúng

$$
\begin{aligned}
(p \land q) \to (p \lor q) &\equiv \neg(p \land q) \lor (p \lor q) && \text{Từ } p \to q \equiv \neg p \lor q \\
&\equiv (\neg p \lor \neg q) \lor (p \lor q) && \text{Luật De Morgan} \\
&\equiv (p \lor \neg p) \lor (q \lor \neg q) && \text{Luật giao hoán, kết hợp} \\
&\equiv \mathbf{T} \lor \mathbf{T} && \text{Luật phủ định} \\
&\equiv \mathbf{T} && \text{Luật nuốt}
\end{aligned}
$$

Bài tập 9

Kiểm tra lại ví dụ trên bằng cách lập bảng chân trị cho $(p \land q) \to (p \lor q)$

Bài tập 10

Ta định nghĩa toán tử `imp1` *thông qua bảng chân trị sau*

p	q	p `imp1` q
T	T	T
T	F	F
F	T	F
F	F	T

Chứng minh rằng p `imp1` $q \equiv q$ `imp1` p.

Bài tập 11

Ta định nghĩa toán tử `imp2` *thông qua bảng chân trị sau*

p	q	$p \texttt{ imp2 } q$
T	T	T
T	F	F
F	T	T
F	F	F

(a) *Chứng minh rằng*

$$(p \texttt{ imp2 } q) \land (q \texttt{ imp2 } p) \not\equiv p \leftrightarrow q$$

(b) *Chứng minh rằng (a) đúng ngay cả khi chúng ta thay đổi hàng thứ ba của bảng chân trị cho* `imp2` *thành* F T F.

Các mệnh đề tương đương
Tương đương lôgic

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

Ví dụ 7

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

- Ứng với *mỗi hàng có giá trị* T ở cột $p \to q$, ta muốn 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	$p \to q$	
T	T	T	$p \wedge q$
T	F	F	
F	T	T	$\neg p \wedge q$
F	F	T	$\neg p \wedge \neg q$

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

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

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)*

Các mệnh đề tương đương
Tương đương lôgic

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

Ví dụ 7

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

- Ứng với *mỗi hàng có giá trị F* ở cột $p \to q$, ta muốn 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	$p \to q$	
T	T	T	
T	F	F	$\neg p \vee q$
F	T	T	
F	F	T	

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

$\neg p \vee q$

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)*

Các mệnh đề tương đương
Tương đương lôgic

- "Khử" $\oplus$, $\rightarrow$, và $\leftrightarrow$ bằng các tương đương lôgic 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 8

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

$p \vee q \rightarrow r \wedge s \equiv \neg(p \vee q) \vee (r \wedge s)$	Khử $\rightarrow$
$\equiv (\neg p \wedge \neg q) \vee (r \wedge s)$	Luật De Morgan
$\equiv (\neg p \vee (r \wedge s)) \wedge (\neg q \vee (r \wedge s))$	Luật phân phối
$\equiv ((\neg p \vee r) \wedge (\neg p \vee s)) \wedge ((\neg q \vee r) \wedge (\neg q \vee s))$	Luật phân phối
$\equiv (\neg p \vee r) \wedge (\neg p \vee s) \wedge (\neg q \vee r) \wedge (\neg q \vee s)$	Luật kết hợp

Bài tập 12

Tìm mệnh đề tương đương chỉ sử dụng các toán tử lôgic trong $\mathcal{C} = \{\neg, \wedge, \vee\}$ *của các mệnh đề*

(1) $p \oplus q$

(2) $p \leftrightarrow q$

Bài tập 13

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

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

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

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

Bài tập 14

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

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

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 (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 $\mathcal{D} = D_1 \times \cdots \times D_n$* nếu $P(a_1, \dots, a_n)$ là một mệnh đề với bộ $(a_1, \dots, a_n)$ bất kỳ trong $\mathcal{D}$ $(a_1 \in D_1, \dots, a_n \in D_n)$

Ví dụ 9

- $P(x) :=$ "x 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 cho y 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) $P(3) \vee P(-1)$

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

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

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

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 vị từ
Lượng từ “với mọi”

- $\forall x\, P(x)$: *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

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

- $\exists x\, P(x)$: *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 lôgic với

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

Ví dụ 10

Mô tả câu "Tất cả sinh viên trong lớp này đã học môn Đại Số" bằng vị từ và lượng từ

- $C(x) :=$ "x đã học môn Đại Số"
- Nếu $\mathcal{D}$ là tập *các sinh viên trong lớp này* $\forall x\, C(x)$
- Nếu $\mathcal{D}$ là tập *tất cả mọi người*. Đặt $S(x) :=$ "x là sinh viên trong lớp này" $\forall x\, (S(x) \rightarrow C(x))$

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

Ví dụ 11

Mô tả câu "Một số sinh viên trong lớp này đã học môn Đại Số" bằng vị từ và lượng từ

- $C(x) :=$ "x đã học môn Đại Số"
- Nếu $\mathcal{D}$ là tập *các sinh viên trong lớp này* $\exists x\, C(x)$
- Nếu $\mathcal{D}$ là tập *tất cả mọi người*. Đặt $S(x) :=$ "x là sinh viên trong lớp này" $\exists x\, (S(x) \wedge C(x))$

Ví dụ 12

Giả sử biến x nhận giá trị từ miền $\mathcal{D}$. Ta chứng minh $\forall x\,(P(x) \wedge Q(x)) \equiv (\forall x\,P(x)) \wedge (\forall x\,Q(x))$. Cụ thể, ta chứng minh hai điều

(1) *Nếu* $\forall x\,(P(x) \wedge Q(x))$ *đúng, thì* $(\forall x\,P(x)) \wedge (\forall x\,Q(x))$ *đúng*
- Giả sử $\forall x\,(P(x) \wedge Q(x))$ đúng. Do đó, với mọi $a \in \mathcal{D}$, $P(a) \wedge Q(a)$ đúng, suy ra $P(a)$ đúng và $Q(a)$ đúng. Do $P(a)$ đúng với mọi $a \in \mathcal{D}$, $\forall x\,P(x)$ đúng. Do $Q(a)$ đúng với mọi $a \in \mathcal{D}$, $\forall x\,Q(x)$ đúng. Do đó $(\forall x\,P(x)) \wedge (\forall x\,Q(x))$ đúng

(2) *Nếu* $(\forall x\,P(x)) \wedge (\forall x\,Q(x))$ *đúng, thì* $\forall x\,(P(x) \wedge Q(x))$ *đúng*
- Giả sử $(\forall x\,P(x)) \wedge (\forall x\,Q(x))$ đúng. Do đó $(\forall x\,P(x))$ đúng và $(\forall x\,Q(x))$ đúng, suy ra với mọi $a \in \mathcal{D}$, $P(a)$ đúng và $Q(a)$ đúng. Như vậy, với mọi $a \in \mathcal{D}$, $P(a) \wedge Q(a)$ đúng. Theo định nghĩa, $\forall x\,(P(x) \wedge Q(x))$ đúng.

Bài tập 16

Chứng minh $\exists x\,(P(x) \vee Q(x)) \equiv (\exists x\,P(x)) \vee (\exists x\,Q(x))$

- 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 $Q(x) := (x > 0) \wedge P(x)$
- 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 $\forall x\, (P(x) \vee Q(x))$

- 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 13

- $P(x, y)$ 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

- *Phủ định* của mệnh đề có lượng từ
  - $\neg\forall x\, P(x) \equiv \exists x\, \neg P(x)$
  - $\neg\exists x\, P(x) \equiv \forall x\, \neg P(x)$
- 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

$$\begin{aligned}\neg\forall x\, P(x) &\equiv \neg(P(x_1) \land P(x_2) \land \cdots \land P(x_n))\\ &\equiv \neg P(x_1) \lor \neg P(x_2) \lor \cdots \lor \neg P(x_n) \quad \text{Luật De Morgan}\\ &\equiv \exists x\, \neg P(x)\\ \neg\exists x\, P(x) &\equiv \neg(P(x_1) \lor P(x_2) \lor \cdots \lor P(x_n))\\ &\equiv \neg P(x_1) \land \neg P(x_2) \land \cdots \land \neg P(x_n) \quad \text{Luật De Morgan}\\ &\equiv \forall x\, \neg P(x)\end{aligned}$$

Ví dụ 14

$P(x) :=$ "x đã học môn Đại Số" với x là một sinh viên trong lớp này

- $\forall x\, P(x) :=$ "Tất cả sinh viên trong lớp này đã học môn Đại Số"
- $\neg\forall x\, P(x) :=$ "Không phải tất cả sinh viên trong lớp này đã học môn Đại Số" $\equiv$ "Ít nhất một sinh viên trong lớp này đã không học môn Đại Số" $=: \exists x\, \neg P(x)$
- $\exists x\, P(x) :=$ "Tồn tại một sinh viên trong lớp này đã học môn Đại Số"
- $\neg\exists x\, P(x) :=$ "Không thể tồn tại một sinh viên trong lớp này đã học môn Đại Số" $\equiv$ "Tất cả sinh viên trong lớp này đã không học môn Đại Số" $=: \forall x\, \neg P(x)$

Ví dụ 15

$P(x, y) :=$ "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 17

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

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

- 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 18

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

Chứng minh

Một số thuật ngữ

- *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

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 $\neg p \to \mathbf{F}$ (phương pháp *Chứng minh phản chứng (Proof by Contradiction)*)

Mục tiêu

Chứng minh $p \to q$ 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) ($\neg q \to \neg p$):** 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 $(p \wedge \neg q) \to \mathbf{F}$)
- **Chứng minh rỗng (vacuous proof):** Chứng minh $\neg p$ đúng mà không cần giả thiết gì khác

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 $\neg p \to \mathbf{F}$

(2) Giả sử tồn tại một số nguyên n vừa chẵn vừa lẻ

(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

Chứng minh sau của Định lý 1

(Với mọi số nguyên n) n không thể vừa chẵn vừa lẻ

đúng hay sai? Tại sao?

Chứng minh phản chứng.

(1) Giả sử tồn tại một số nguyên n vừa chẵn vừa lẻ
(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

$\square$

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

$\square$

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 $\neg q \rightarrow \neg p$
- *(Mệnh đề phản đảo: Nếu n chẵn, thì $3n + 2$ cũng chẵn)*

(1) Giả sử kết luận của định lý trên là sai, nghĩa là 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, và do đó là số chẵn

(5) Ta đã chứng minh $\neg$*(n lẻ)* $\rightarrow \neg$*($3n + 2$ lẻ)* đúng, do đó mệnh đề phản đảo *($3n + 2$ lẻ)* $\rightarrow$ *(n lẻ)* cũng đúng

□

Chứng minh
Ví dụ

Định lý 4

(Với mọi số nguyên n) Nếu n vừa chẵn vừa lẻ, thì $n^2 = n + n$

Chứng minh rỗng.

- **Nhắc lại:** để chứng minh $p \to q$, ta chứng minh $\neg p$ mà không cần bất cứ giả thiết nào

(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)

□

Định lý 5

(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 $p \to q$, ta chứng minh q mà không cần bất cứ giả thiết nào

(1) Với mọi số nguyên n, mệnh đề "hoặc n chẵn hoặc n lẻ" đúng

(2) Do đó, kết luận của mệnh đề cần chứng minh luôn đúng, bất luận giả thiết là đúng hay sai

(3) Hiển nhiên là mệnh đề cần chứng minh luôn đúng

□

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) $a = b$ Giả thiết

(2) $a^2 = ab$ Nhân hai vế của (1) với a

(3) $a^2 - b^2 = ab - b^2$ Trừ b^2 từ cả hai vế của (2)

(4) $(a - b)(a + b) = (a - b)b$ Phân tích hai vế của (3) thành nhân tử

(5) $a + b = b$ Chia cả hai vế của (4) cho $a - b$

(6) $2b = b$ Thay a bởi b trong (5) vì $a = b$, và đơn giản hóa

(7) $2 = 1$ Chia cả hai vế của (6) cho b

Chứng minh sau của mệnh đề

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

là đúng hay 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

□

Part I

Phụ lục

Các quy tắc suy luận
Giới thiệu

- *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

- Một lập luận lôgic được gọi là *hợp lý (valid)* nếu *khi mọi giả thiết đúng thì kết luận cũng đúng*. Một lập luận lôgic 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 lôgic 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)

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

$$((\text{Giả thiết 1}) \land (\text{Giả thiết 2}) \land \cdots) \to (\text{Kết luận})$$

Ví dụ 16

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

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

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

$$\begin{array}{rl} & p \\ & p \to q \\ \hline \therefore & q \end{array}$$

$$(p \wedge (p \to q)) \to q$$

Modus Ponens — Hằng đúng tương ứng

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

Ví dụ 17

Nếu $\left\{ \begin{array}{ll} p \to q & \text{"Nếu } n \text{ chia hết cho } 3, \text{ thì } n^2 \text{ chia hết cho } 3\text{"} \\ p & \text{"} n \text{ chia hết cho } 3\text{"} \end{array} \right\}$ là ĐÚNG

thì $\therefore\ q$ "n^2 chia hết cho 3" cũng ĐÚNG

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

$$\boxed{\begin{array}{rl} & \neg q \\ & p \to q \\ \hline \therefore & \neg p \end{array}}$$

Modus Tollens

$$\boxed{(\neg q \land (p \to q)) \to \neg p}$$

Hằng đúng tương ứng

Ví dụ 18

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

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

- Quy tắc *Cộng (Addition)*

$$\begin{array}{rl} & p \\ \hline \therefore & p \vee q \end{array}$$

Quy tắc Cộng (Addition)

$$\boxed{p \rightarrow (p \vee q)}$$

Hằng đúng tương ứng

- Quy tắc *Rút gọn (Simplification)*

$$\begin{array}{rl} & p \wedge q \\ \hline \therefore & p \end{array}$$

Quy tắc Rút gọn (Simplification)

$$\boxed{(p \wedge q) \rightarrow p}$$

Hằng đúng tương ứng

- Quy tắc *Hội (Conjunction)*

$$\begin{array}{rl} & p \\ & q \\ \hline \therefore & p \wedge q \end{array}$$

Quy tắc Hội (Conjunction)

$$\boxed{((p) \wedge (q)) \rightarrow p \wedge q}$$

Hằng đúng tương ứng

Ví dụ 19

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 $0°C$. Do đó, nhiệt độ hiện tại là dưới $0°C$ hoặc trời đang mưa [Quy tắc Cộng]

(c) Nhiệt độ hiện tại là dưới $0°C$ và trời đang mưa. Do đó, nhiệt độ hiện tại là dưới $0°C$ [Quy tắc Rút gọn]

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

$$\begin{array}{rl} & p \to q \\ & q \to r \\ \hline \therefore & p \to r \end{array}$$

Tam đoạn luận giả định (Hypothetical syllogism)

$$((p \to q) \land (q \to r)) \to (p \to r)$$

Hằng đúng tương ứng

Ví dụ 20

"Nếu $\underbrace{\text{trời mưa hôm nay}}_{p}$, thì $\underbrace{\text{chúng ta sẽ không tổ chức tiệc nướng hôm nay}}_{q}$. Nếu $\underbrace{\text{chúng ta không tổ chức tiệc nướng hôm nay}}_{q}$, thì $\underbrace{\text{chúng ta sẽ tổ chức tiệc nướng vào ngày mai}}_{r}$. Do đó nếu $\underbrace{\text{trời mưa hôm nay}}_{p}$, thì $\underbrace{\text{chúng ta sẽ tổ chức tiệc nướng vào ngày mai}}_{r}$."

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

$$\boxed{\begin{array}{rl} & p \lor q \\ & \neg p \\ \hline \therefore & q \end{array}}$$

Tam đoạn luận tuyển (Disjunctive syllogism)

$$\boxed{((p \lor q) \land (\neg p)) \to q}$$

Hằng đúng tương ứng

Ví dụ 21

"$\underbrace{\text{Ví của tôi nằm trong túi áo khoác}}_{p}$ hoặc $\underbrace{\text{nó nằm trên bàn}}_{q}$. $\underbrace{\text{Ví của tôi không nằm trong túi áo khoác}}_{\neg p}$. Do đó, $\underbrace{\text{nó nằm trên bàn}}_{q}$."

Các quy tắc suy luận

Một số quy tắc suy luận trong lôgic mệnh đề

- Quy tắc *Hợp giải (Resolution)*

$$\begin{array}{rl} & p \vee q \\ & \neg p \vee r \\ \hline \therefore & q \vee r \end{array} \qquad ((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$$

Hợp giải (Resolution) — Hằng đúng tương ứng

- 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 tuyển*

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

Ví dụ 22

"$\underbrace{\text{Tôi đang đi trên đường}}_{q}$ hoặc $\underbrace{\text{trời mưa}}_{p}$. $\underbrace{\text{Trời không mưa}}_{\neg p}$ hoặc $\underbrace{\text{tôi đang ở nhà}}_{r}$. Do đó, $\underbrace{\text{tôi đang đi trên đường}}_{q}$ hoặc $\underbrace{\text{tôi đang ở nhà}}_{r}$."

- Cho trước các tiền đề (premises) $p_1, p_2, \ldots, p_n$. 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 23

- 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

- **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	$\neg$*sunny* $\land$ *cold*	Trời không nắng và thời tiết lạnh
p_2	*swim* $\to$ *sunny*	Chúng ta sẽ đi bơi chỉ khi trời nắng
p_3	$\neg$*swim* $\to$ *canoe*	Nếu chúng ta không đi bơi, thì chúng ta sẽ đi chèo xuồng
p_4	*canoe* $\to$ *sunset*	Nếu chúng ta đi chèo xuống, chúng ta sẽ về đến nhà lúc hoàng hôn
$\therefore$	*sunset*	Chúng ta sẽ về đến nhà lúc hoàng hôn

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

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

$$
\begin{array}{ll}
p_1 & \neg sunny \wedge cold \\
p_2 & swim \rightarrow sunny \\
p_3 & \neg swim \rightarrow canoe \\
p_4 & canoe \rightarrow sunset \\
\hline
\therefore & sunset
\end{array}
$$

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

Ví dụ 24

- Nếu David Cameron là Tổng thống Hoa Kỳ, thì ông ta ít nhất là bốn mươi tuổi $(p \to q)$
- David Cameron ít nhất là bốn mươi tuổi (q)
- Do đó, David Cameron là Tổng thống Hoa Kỳ (p)

- *Ngụy biện phủ định giả thiết (Fallacy of denying the hypothesis)*
  - *$p \to q$ là đúng, và p sai. Do đó q là sai* (Sai. Bởi vì $\mathbf{F} \to \mathbf{T}$ đúng)

Ví dụ 25

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

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

$$\begin{array}{rl} & \forall x\, P(x) \\ \hline \therefore & P(c) \end{array}$$

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

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

$$\begin{array}{rl} & P(c) \\ \hline \therefore & \forall x\, P(x) \end{array}$$

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

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

$$\begin{array}{rl} & \exists x\, P(x) \\ \hline \therefore & P(c) \end{array}$$

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

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

$$\begin{array}{rl} & P(c) \\ \hline \therefore & \exists x\, P(x) \end{array}$$

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

Ví dụ 26

- 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 là người"
  - $D(x) =$ "x sẽ chết"
  - $S =$ "Socrates" – một phần tử trong vũ trụ
- **Xác định lập luận**

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

Ví dụ 27

- 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**

p_1	$\exists x\,(S(x) \wedge I(x))$	Ít nhất một trong số các sinh viên trong lớp rất thông minh
p_2	$S(J)$	John là một sinh viên trong lớp
$\therefore$	$I(J)$	John rất thông minh

- **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 $p_1 = \exists x\,(S(x) \cap I(x))$ 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 Khởi tạo phổ quát cho ta *kết luận $I(J)$ là sai*

Bài tập 19

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 20

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*

Bài tập 21

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

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?

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]".

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

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)

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)

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.

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

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)

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 vẫy 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)

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 trông 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 chùm[1]. 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.)

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

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

VNU-HUS MAT3500: Toán rời rạc

Các cấu trúc cơ bản

Tập hợp, Hàm, Dãy, Tổng

Hoàng Anh Đức

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

Nội dung

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

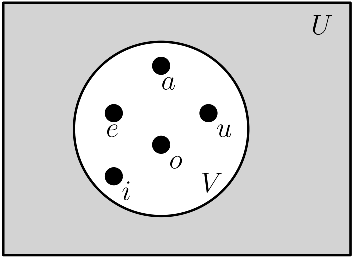
Tổng
- Ký hiệu tổng và một số khái niệm
- Một số công thức tổng hữu ích

- 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 tập hợp thông qua *giản đồ Venn (Venn diagram)*
  - *Tập vũ trụ (universal set)* U gồm tất cả các đối tượng đang xét — Hình chữ nhật
  - Tập hợp cần mô tả — Hình tròn hoặc các hình khác
  - Phần tử của tập hợp — Điểm



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

- *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

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

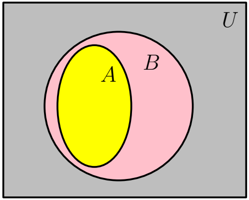
- $(A \subseteq B) \equiv \forall x\,(x \in A \to x \in B)$
- $(A \nsubseteq B) \equiv \neg(A \subseteq B)$ (A *không* là tập con của B)
- $(A \subset B) \equiv (A \subseteq B) \wedge (B \nsubseteq A)$ (A là *tập con thực sự (proper subset)* của B)

Bài tập 1

Chứng minh các mệnh đề sau

(1) *Nếu $A \subseteq B$ và $B \subseteq C$ thì $A \subseteq C$*

(2) *Với mọi tập A, ta có $\emptyset \subseteq A$ và $A \subseteq A$*



Hình: Giản đồ Venn mô tả $A \subset B$

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, $\{a, b, c\} = \{a, c, b\} = \{b, a, c\} = \{b, c, a\} = \{c, a, b\} = \{c, b, a\}$

Bài tập 2

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

Bài tập 3

Các mệnh đề sau đúng hay sai?

(a) $1 \in \{1\}$

(b) $1 \subseteq \{1\}$

(c) $\{1\} \in \{\{1\}\}$

(d) $\{1\} \subseteq \{\{1\}\}$

Bài tập 4

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

- *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 (**n**atural 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)*
  - $\mathbb{Q} = \{p/q \mid p, q \in \mathbb{Z},$ và $q \neq 0\}$ *Tập số hữu tỷ (rational numbers)*
  - $\mathbb{R}$ *Tập số thực (**r**eal numbers)*
  - $\mathbb{R}^+$ *Tập số thực dương (positive real numbers)*
  - $\mathbb{C}$ *Tập số phức (**c**omplex numbers)*
- $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$

- *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 5

*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:** $A = B \equiv (A \subseteq B) \land (B \subseteq A) \equiv \forall x\,(x \in A \leftrightarrow x \in B)$ và nếu $A \subseteq B$ và $B \subseteq C$ thì $A \subseteq C$)*

- 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, ..., 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 Đềcác (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 $b \in B$
  - $A \times B = \{(a, b) \mid a \in A \land b \in B\}$
  - Chú ý rằng tích Đềcác *không* có tính chất giao hoán, nghĩa là $\neg\forall A, B\ (A \times B = B \times A)$
  - Tổng quát hóa $A_1 \times \cdots \times A_n = \{(a_1, \dots, a_n) \mid a_1 \in A_1 \land \cdots \land a_n \in A_n\}$

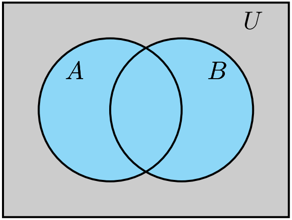
Bài tập 6

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

Bài tập 7

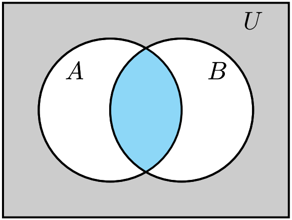
Chứng minh rằng $A \times B = B \times A$ *khi và chỉ khi* $A = \emptyset$ *hoặc* $B = \emptyset$ *hoặc* $A = B$

- *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: Giản đồ Venn mô tả $A \cup B$

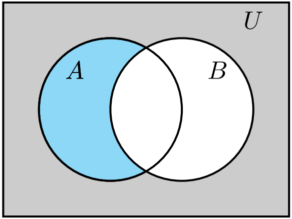
- *Giao (intersection)* của hai tập hợp A, B, ký hiệu $A \cap B$, là tập chứa tất cả các phần tử đồng thời thuộc cả A và B
  - $\forall A, B\, (A \cap B = \{x \mid x \in A \land x \in B\})$
  - $A \cap B \subseteq A$ và $A \cap B \subseteq B$
  - $\{1, 3, 5\} \cap \{2, 3, 4\} = \{3\}$
- Hai tập A và B là *rời nhau (disjoint)* nếu $A \cap B = \emptyset$.
  - $\{1, 3, 5\} \cap \{2, 4, 6\} = \emptyset$



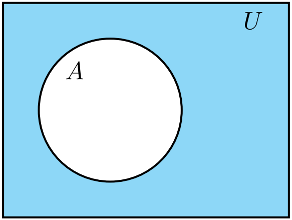
Hình: Giản đồ Venn mô tả $A \cap B$

Tập hợp
Phép hiệu

- *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 $\overline{A}$, là tập $U - A$
  - $\forall A\, (\overline{A} = \{x \mid x \notin A\})$

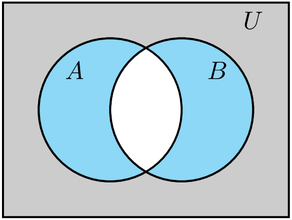


Hình: Giản đồ Venn mô tả $A - B$



Hình: Giản đồ Venn mô tả $\overline{A}$

- *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: Giản đồ Venn mô tả $A \Delta B$

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 8

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
Các hằng đẳng thức tập hợp

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)	$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$
Luật phân phối (Distributive laws)	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

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 lôgic
(3) Chứng minh bằng bảng tính thuộc
(4) Chứng minh bằng giản đồ Venn

Ví dụ 1 (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 $x \in \overline{A \cap B}$

Ví dụ 2 (Dùng đẳng thức lôgic đã biết)

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

$\overline{A \cap B} = \{x \mid x \notin A \cap B\}$	định nghĩa phần bù
$= \{x \mid \neg(x \in A \cap B)\}$	định nghĩa $\notin$
$= \{x \mid \neg(x \in A \land x \in B)\}$	định nghĩa $\cap$
$= \{x \mid \neg(x \in A) \lor \neg(x \in B)\}$	luật De Morgan
$= \{x \mid x \notin A \lor x \notin B\}$	định nghĩa $\notin$
$= \{x \mid x \in \overline{A} \lor x \in \overline{B}\}$	định nghĩa phần bù
$= \{x \mid x \in \overline{A} \cup \overline{B}\}$	định nghĩa $\cup$
$= \overline{A} \cup \overline{B}$	mô tả tập hợp

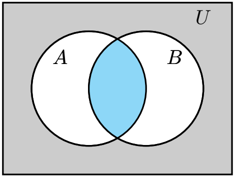
Ví dụ 3 (Dùng bảng tính thuộc)

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

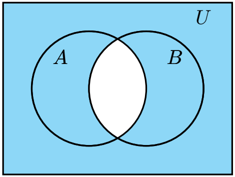
A	B	$\overline{A}$	$\overline{B}$	$A \cap B$	$\overline{A \cap B}$	$\overline{A} \cup \overline{B}$
1	1	0	0	1	0	0
1	0	0	1	0	1	1
0	1	1	0	0	1	1
0	0	1	1	0	1	1

Ví dụ 4 (Dùng giản đồ Venn)

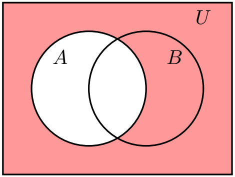
Chứng minh $\overline{A \cap B} = \overline{A} \cup \overline{B}$



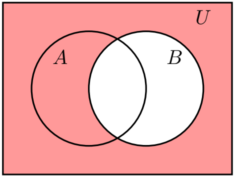
Hình: Giản đồ Venn mô tả $A \cap B$



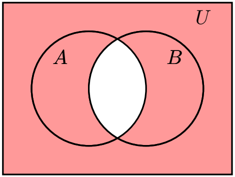
Hình: Giản đồ Venn mô tả $\overline{A \cap B}$



Hình: Giản đồ Venn mô tả $\overline{A}$



Hình: Giản đồ Venn mô tả $\overline{B}$



Hình: Giản đồ Venn mô tả $\overline{A} \cup \overline{B}$

Bài tập 9

Chứng minh các hằng đẳng thức tập hợp đã đề cập (sử dụng các phương pháp đã ví dụ ở trên)

Bài tập 10

Với hai tập A, B bất kỳ, chứng minh

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

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

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

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

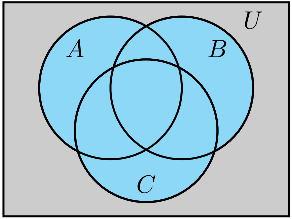
(e) $A \Delta A = \emptyset$

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

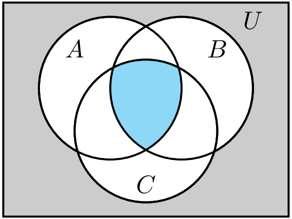
(g) $A \Delta B = B \Delta A$

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

- 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: Giản đồ Venn cho $A \cup B \cup C$



Hình: Giản đồ Venn cho $A \cap B \cap C$

- *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 tập hợp $\bigcup_{i=1}^{\infty} A_i$
  - Ví dụ, với $i = 1, 2, \dots$ nếu $A_i = \{i, i+1, i+2, \dots\}$ thì $\bigcup_{i=1}^{n} A_i = \bigcup_{i=1}^{n} \{i, i+1, i+2, \dots\} = \{1, 2, 3, \dots\} = \mathbb{Z}^+$

Tập hợp

Tổng quát hóa phép hợp và phép giao

- *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
  - $\bigcap_{i=1}^{n} A_i = \{x \mid \forall i \in \{1, \dots, n\}\, (x \in A_i)\} = A_1 \cap A_2 \cap \dots \cap A_n$
  - Tương tự với tập chỉ số I bất kỳ $\bigcap_{i \in I} A_i$ hay với vô hạn các tập hợp $\bigcap_{i=1}^{\infty} A_i$
  - Ví dụ, với $i = 1, 2, \dots$ nếu $A_i = \{i, i+1, i+2, \dots\}$ thì $\bigcap_{i=1}^{n} A_i = \bigcap_{i=1}^{n} \{i, i+1, i+2, \dots\} = \{n, n+1, n+2, \dots\} = A_n$

- 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 $\mathcal{B}(A) = x_1 x_2 \dots x_n$ 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 $\mathcal{B}(A) = 0110101000$

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

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

Bài tập 11

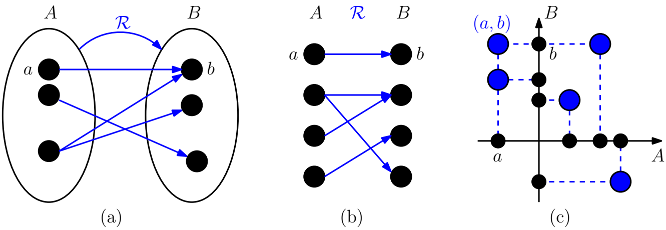
Với $U = \{1, 2, \dots, 10\}$ ($u_i = i$), $A_1 = \{2, 3, 5, 7\}$, $A_2 = \{1, 3, 9\}$, hãy so sánh

(1) $\mathcal{B}(A_1 \cup A_2)$ *và* $\mathcal{B}(A_1) \vee \mathcal{B}(A_2)$

(2) $\mathcal{B}(A_1 \cap A_2)$ *và* $\mathcal{B}(A_1) \wedge \mathcal{B}(A_2)$

(3) $\mathcal{B}(\overline{A_1})$ *và* $\neg\mathcal{B}(A_1)$

- Cho hai tập hợp A và B. Một *quan hệ (relation)* $\mathcal{R}$ giữa A và B là một tập con của tích Đềcác $A \times B$. 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 Đềcác

- 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 12

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 lôgic*

(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*

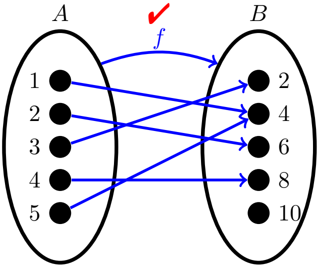
(4) $\mathcal{R} = \{(a, b) \mid b$ *chia hết cho* $a\}$ *với* a, b *là các số nguyên dương*

Hàm

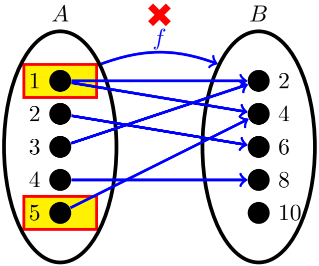
Định nghĩa hàm và một số khái niệm

- Với hai tập khác rỗng A, B, một *hàm (function)* f từ A đến B, ký hiệu $f : A \to B$, 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

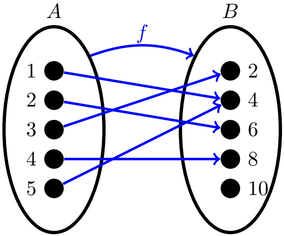
Giả sử f là một hàm từ A đến B

- 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 *nghịch ảnh (preimage)* của b. Tập hợp tất cả các ảnh của các phần tử thuộc A được gọi là *ảnh của A* qua hàm f, ký hiệu $f(A)$
    - $f(A) \subseteq B$
- Ta cũng nói rằng f *ánh xạ* A đến B

Ví dụ 5

Với hàm f như hình bên

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



Hình: $f : A \to B$

- 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

$$\boxed{(f_1 + f_2)}(x) = \boxed{f_1(x) + f_2(x)}$$
$$\boxed{(f_1 f_2)}(x) = \boxed{f_1(x) f_2(x)}$$

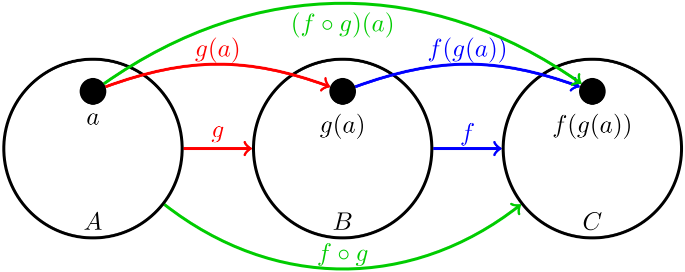
phép toán trong $\mathbb{R}$

Bài tập 13

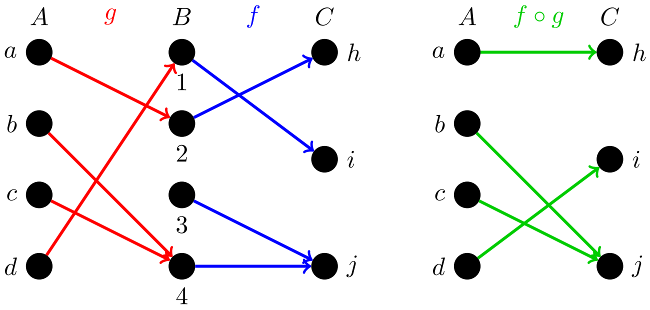
Hãy kiểm tra lại rằng $f_1 + f_2$ và $f_1 f_2$ thực sự là các hàm số

- Giả sử f là hàm số từ A đến B. Có thể mở rộng định nghĩa ảnh của tập xác định A cho một tập con S của nó. *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

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



Ví dụ 6

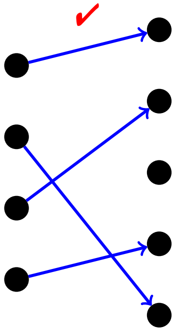


Bài tập 14

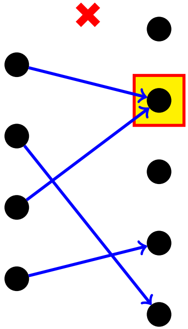
Cho $g : \{a, b, c\} \to \{a, b, c\}$ *với* $g(a) = b$, $g(b) = c$, *và* $g(c) = a$.
Cho $f : \{a, b, c\} \to \{1, 2, 3\}$ *với* $f(a) = 3$, $f(b) = 2$, *và* $f(c) = 1$.
Hãy tìm $f \circ g$ *và* $g \circ f$

- Hàm $f : A \to B$ 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
  - $\forall a, b\,(f(a) = f(b) \to a = b) \equiv \forall a, b\,(a \neq b \to f(a) \neq f(b))$

Ví dụ 7



Hình: Đơn ánh



Hình: Không phải đơn ánh

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

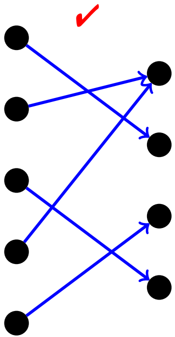
- f được gọi là *tăng (increasing)* khi và chỉ khi với mọi x, y thuộc A thỏa mãn $x < y$, ta luôn có $f(x) \leq f(y)$
  - $\forall x \forall y\, (x < y \to f(x) \leq f(y))$
- f được gọi là *thực sự tăng (strictly increasing)* khi và chỉ khi với mọi x, y thuộc A thỏa mãn $x < y$, ta luôn có $f(x) < f(y)$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

Bài tập 15

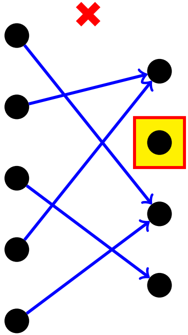
Chứng minh nếu f là hàm thực sự tăng hoặc thực sự giảm thì f là đơn ánh

- Hàm $f : A \to B$ 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 8



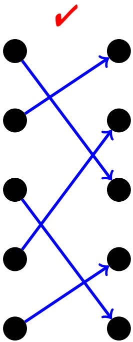
Hình: Toàn ánh



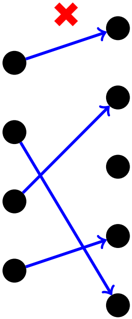
Hình: Không phải toàn ánh

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

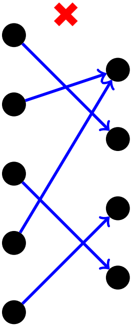
Ví dụ 9



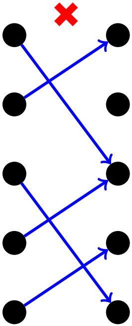
Hình: Song ánh



Hình: Đơn ánh, không toàn ánh

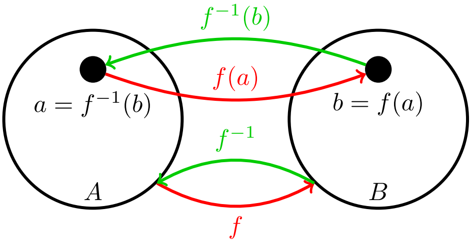


Hình: Toàn ánh, không đơn ánh



Hình: Không đơn ánh, không toàn ánh

- Cho $f : A \to B$ 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 $f^{-1} : B \to A$
- Một song ánh còn được gọi là một *hàm khả nghịch (invertible function)*



Bài tập 16

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

Bài tập 17

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

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

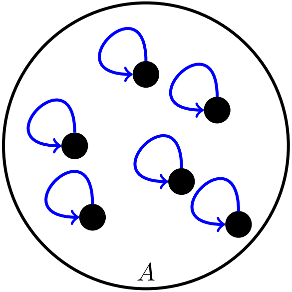
(b) $f : \mathbb{R} \to \mathbb{R}, f(x) = x^2$

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

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

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

$$f^{-1} \circ f = id_A$$

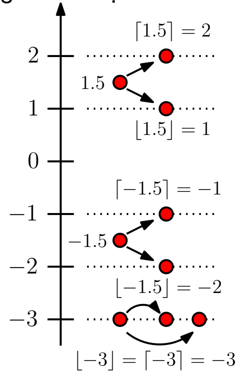


Hình: Hàm đồng nhất trên A

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 10

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

Dãy

Định nghĩa dãy và một số khái niệm

- Một *dãy (sequence)* $\{a_n\}$ được xác định qua một hàm $f : I \to A$ trong đó $I \subseteq \mathbb{Z}$ và A là tập bất kỳ
  - Thông thường, $I = \mathbb{N}$ hoặc $I = \mathbb{Z}^+ = \mathbb{N} - \{0\}$
  - Ví dụ, dãy $\{a_n\}$ xác định bởi $f(n) = n^2$ với mọi số nguyên $n \geq 0$ có các phần tử $0, 1, 4, 9, 16, \dots$
- Với $n \in I$, ta *sử dụng a_n để chỉ ảnh của n*, nghĩa là $a_n = f(n)$.
  - a_n là một *số hạng (term)* của dãy $\{a_n\}$
  - n là *chỉ số (index)* của a_n (thông thường, ta sử dụng i thay vì n)
- Đôi khi, thay vì ký hiệu $\{a_n\}$, có thể viết *"dãy $a_1, a_2, \dots$"* để chắc chắn rằng tập các chỉ số I được xác định rõ ràng
- Có thể mô tả một dãy bằng cách liệt kê một vài phần tử đầu tiên hoặc cuối cùng của dãy và sử dụng "..." cho phần còn lại
  - Ví dụ, có thể mô tả dãy $\{a_n\}$ ở trên bằng cách viết $\{a_n\} = 0, 1, 4, 9, 16, 25, \dots$

- 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

Dãy

Dãy cho bởi hệ thức truy hồi

- 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 *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 trong 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 $a_n = n^2$ $(n \geq 0)$

Ví dụ 11

- 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 $n \geq 1$
  - $\{g_n\} = 1, 1, 2, 6, 24, \dots$

Ví dụ 12

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 $d_n = -1 + 4n$

Ví dụ 13

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 $a_n =$

Dãy
Tìm công thức tường minh của một dãy

- 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 14
Số tiếp theo trong dãy có thể là bao nhiêu?

- $1, 2, 3, 4, \ldots$
- $1, 3, 5, 7, 9, \ldots$
- $2, 3, 5, 7, 11, \ldots$

Ví dụ 15
Các số hạng tiếp theo có thể là bao nhiêu?

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

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

- Bảng Tra Cứu Dãy Số Nguyên Trực Tuyến (The On-Line Encyclopedia of Integer Sequences - OEIS) `https://oeis.org/`

- 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

$$a_m + a_{m+1} + \cdots + a_n$$

$$\sum_{j=m}^{n} a_j$$

$$\sum_{m \leq j \leq n} a_j$$

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

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

- Với tập chỉ số S bất kỳ, ta có thể viết

$$\sum_{j \in S} a_j$$

- Với $\{a_n\}$ là dãy vô hạn, ta có thể viết

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

- Tổng các giá trị của một hàm trên tập $X = \{x_1, x_2, \dots\}$

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

- Nếu $X = \{x \mid P(x)\}$ với vị từ $P(x)$ nào đó

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

Tổng

Ký hiệu tổng và một số khái niệm

Ví dụ 16

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

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

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

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

Tổng

Một số công thức tổng hữu ích

- *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,

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

- *Giao hoán:*

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

- *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:*

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

- 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

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

$$\begin{aligned} S &= a + ar + ar^2 + ar^3 + \cdots + ar^n \\ rS &= \phantom{a +{}} ar + ar^2 + ar^3 + \cdots + ar^n + ar^{n+1} \\ rS - S &= ar^{n+1} - a \end{aligned}$$

Tổng

Một số công thức tổng hữu ích

$$
\begin{aligned}
rS &= r\sum_{i=0}^{n} ar^i && \text{công thức của } S\\
&= \sum_{i=0}^{n} ar^{i+1} && \text{phân phối}\\
&= \sum_{k=1}^{n+1} ar^k && \text{đổi chỉ số, } k = i+1\\
&= \sum_{k=1}^{n} ar^k + \sum_{k=n+1}^{n+1} ar^k && \text{tách tổng}\\
&= (\sum_{k=1}^{n} ar^k + ar^0) + (ar^{n+1} - ar^0) && \text{thêm và bớt } ar^0 = a\\
&= \sum_{k=0}^{n} ar^k + (ar^{n+1} - a) && \text{tách tổng}\\
&= S + (ar^{n+1} - a) && \text{công thức của } S
\end{aligned}
$$

Ví dụ 17

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

$$
\begin{aligned}
T &= 1 + 2 + 3 + \cdots + n \\
T &= n + (n-1) + (n-2) + \cdots + 1 \\
2T &= (n+1) \cdot n
\end{aligned}
$$

Bài tập 18

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

Ví dụ 18

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

$$1 + 2 + 3 + 4 + \ldots + 97 + 98 + 99 + 100$$

Bài tập 19

Tìm công thức tường minh cho $T = \sum_{i=1}^{n} i$ sử dụng phương pháp tương tự như ví dụ trên. Có thể áp dụng phương pháp tương tự cho Bài tập 18 không?

Ví dụ 19

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

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

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

Part I

Phụ lục

Nội dung

- Chúng ta đang học một *lý thyết 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 lôgic 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 $S = \{A \mid A$ 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 $S \in S$?

- **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 20

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 21

Tập $2\mathbb{Z}$ gồm các số nguyên chẵn có cùng lực lượng với tập số nguyên $\mathbb{Z}$ hay không?

Định lý 1: Định lý Cantor

Không tồn tại một toàn ánh $f : A \to \mathcal{P}(A)$ 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 $f : A \to \mathcal{P}(A)$
- 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 $a \notin f(a)$. Do đó theo định nghĩa của G, ta có $a \in G$. Đây là một mâu thuẫn

Lực lượng của tập vô hạn

Tập đếm được và không đếm được

- 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 20

Tập các số tự nhiên $\mathbb{N}$ là tập đếm được

$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & \cdots \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \\ 1 & 2 & 3 & 4 & 5 & \cdots \end{array}$$

Ví dụ 21

Tập các số nguyên dương lẻ là tập đếm được

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & \cdots \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \\ 1 & 3 & 5 & 7 & 9 & \cdots \end{array}$$

Ví dụ 22

Tập các số nguyên $\mathbb{Z}$ là tập đếm được

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & \cdots \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \\ 0 & 1 & -1 & 2 & -2 & \cdots \end{array}$$

Ví dụ 23

Tập các số hữu tỷ dương $\mathbb{Q}^+ = \{\frac{p}{q} \mid p, q \in \mathbb{Z}^+\}$ là tập đếm được

$$\begin{array}{cccccc} \boxed{\frac{1}{1}} & \boxed{\frac{2}{1}} & \boxed{\frac{3}{1}} & \boxed{\frac{4}{1}} & \boxed{\frac{5}{1}} & \cdots \\ \boxed{\frac{1}{2}} & \frac{2}{2} & \boxed{\frac{3}{2}} & \frac{4}{2} & \frac{5}{2} & \cdots \\ \boxed{\frac{1}{3}} & \boxed{\frac{2}{3}} & \frac{3}{3} & \frac{4}{3} & \frac{5}{3} & \cdots \\ \boxed{\frac{1}{4}} & \frac{2}{4} & \frac{3}{4} & \frac{4}{4} & \frac{5}{4} & \cdots \\ \boxed{\frac{1}{5}} & \frac{2}{5} & \frac{3}{5} & \frac{4}{5} & \frac{5}{5} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{array}$$

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & \cdots \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \\ \frac{1}{1} & \frac{1}{2} & \frac{2}{1} & \frac{3}{1} & \frac{1}{3} & \cdots \end{array}$$

Lực lượng của tập vô hạn

Tập đếm được và không đếm được

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: $r_1, r_2, \dots$

$$
\begin{aligned}
r_1 &= 0.d_{11}d_{12}d_{13}d_{14}\dots\\
r_2 &= 0.d_{21}d_{22}d_{23}d_{24}\dots\\
r_3 &= 0.d_{31}d_{32}d_{33}d_{34}\dots\\
r_4 &= 0.d_{41}d_{42}d_{43}d_{44}\dots\\
&\vdots
\end{aligned}
$$

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

- 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

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 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

- *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
  - độ phức tạp của thuật toán
  - tính đúng đắn của một số chương trình máy tính
  - các định lý về đồ thị và cây
- 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

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 $P(k) \to P(k+1)$ 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 lôgic,

$$(P(1) \land \forall k \in \mathbb{Z}^+ \; (P(k) \to P(k+1))) \to \forall n \in \mathbb{Z}^+ \; P(n)$$

- 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 $P(k) \to P(k+1)$ đúng với mọi $k \in \mathbb{Z}^+$. Theo định nghĩa của toán tử lôgic "$\to$", 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)

Một số cách minh họa phương pháp quy nạp

- *Bí mật mọi người biết:* Ta giả sử mọi người xếp thành một hàng dài vô tận và được đánh số $1, 2, \ldots, k, \ldots$ Để thuyết phục ai đó rằng mọi người đều biết một bí mật, bạn có thể lý luận rằng
  - Người thứ nhất biết bí mật
  - Nếu người thứ k biết bí mật, thì anh/cô ấy nói cho người thứ $k + 1$ bí mật đó
- *Hiệu ứng đôminô:* Ta có một hàng dài vô tận các quân bài đôminô đánh số $1, 2, \ldots, k, \ldots$ Để thuyết phục ai đó rằng quân bài thứ k sẽ đổ, bạn có thể lý luận rằng
  - Quân bài thứ nhất đổ, và
  - Bất kể khi nào một quân bài đổ, quân bài ngay tiếp theo nó cũng đổ

Chứng minh (Quy nạp yếu là đúng).

- Giả sử $P(1)$ đúng và với mọi $k \in \mathbb{Z}^+$, $P(k) \to P(k+1)$ 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 $P(k-1) \to P(k)$ đúng với mọi $k \in \mathbb{Z}^+$, ta có $P(m-1) \to P(m)$ 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

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 $P(k) \to P(k+1)$ đúng với mọi số nguyên $k \geq b$

Theo ngôn ngữ lôgic,

$$(P(b) \land \forall k \in \mathbb{Z}^{\geq b}\, (P(k) \to P(k+1))) \to \forall n \in \mathbb{Z}^{\geq b}\, P(n),$$

trong đó $\mathbb{Z}^{\geq b} = \{m \mid m \in \mathbb{Z} \text{ và } m \geq b\}$ (chú ý là $\mathbb{Z}^{\geq 1} = \mathbb{Z}^{+}$)

Bài tập 1

*Chứng minh nguyên lý quy nạp yếu tổng quát là đúng. (**Gợi ý:** Đặt $Q(n-b+1) = P(n)$. Ta có $\forall n \geq b\, P(n) \equiv \forall n \geq 1\, Q(n)$)*

(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 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

Ví dụ 1

Ta chứng minh phát biểu $P(n)$ sau đúng với mọi $n \in \mathbb{Z}^+$

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

- **Bước cơ sở.** $P(1)$ đúng, do vế trái bằng 1 và vế phải bằng $\frac{1(1+1)}{2} = 1$
- **Bước quy nạp.** Giả sử $P(k)$ đúng với một số nguyên cố định $k \geq 1$ nào đó, nghĩa là $\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$. Ta chứng minh $P(k+1)$ đúng, nghĩa là $\sum_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2}$. Ta có

$$\begin{aligned}
\sum_{i=1}^{k+1} i &= \sum_{i=1}^{k} i + (k+1) && \text{tách tổng} \\
&= \frac{k(k+1)}{2} + (k+1) && \text{giả thiết quy nạp} \\
&= \frac{(k+1)(k+2)}{2}
\end{aligned}$$

Ví dụ 2

Ta chứng minh phát biểu $P(n)$ sau đúng với mọi $n \in \mathbb{N}$

$$\sum_{i=0}^{n} r^i = \frac{r^{n+1} - 1}{r - 1} \text{ với số thực } r \neq 1 \text{ bất kỳ cho trước}$$

- **Bước cơ sở.** $P(0)$ đúng, do vế trái bằng 1 và vế phải bằng $\frac{r^1-1}{r-1} = 1$
- **Bước quy nạp.** Giả sử $P(k)$ đúng với một số nguyên cố định $k \geq 0$ nào đó, nghĩa là $\sum_{i=0}^{k} r^i = \frac{r^{k+1}-1}{r-1}$. Ta chứng minh $P(k+1)$ đúng, nghĩa là $\sum_{i=0}^{k+1} r^i = \frac{r^{k+2}-1}{r-1}$

$$\begin{aligned}
\sum_{i=0}^{k+1} r^i &= \sum_{i=0}^{k} r^i + r^{k+1} && \text{tách tổng} \\
&= \frac{r^{k+1} - 1}{r - 1} + r^{k+1} && \text{giả thiết quy nạp} \\
&= \frac{r^{k+2} - 1}{r - 1}
\end{aligned}$$

Ví dụ 3

Ta chứng minh phát biểu $Q(n)$ sau đúng với mọi $n \in \mathbb{N}$

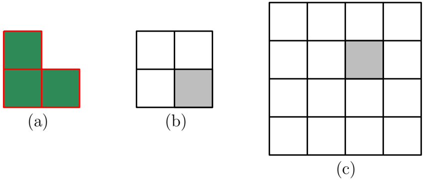
$$|\mathcal{P}(A)| = 2^n \text{ với mọi tập hợp } A \text{ thỏa mãn } |A| = n$$

- **Bước cơ sở.** $Q(0)$ đúng ví $A = \emptyset$ là tập duy nhất thỏa mãn $|A| = 0$ và ta có vế trái $|\mathcal{P}(\emptyset)| = |\{\emptyset\}| = 1$ bằng với vế phải $2^0 = 1$
- **Bước quy nạp.** Giả sử $Q(k)$ đúng với một số nguyên cố định $k \geq 0$ nào đó, nghĩa là $|\mathcal{P}(A)| = 2^k$ với mọi tập hợp A thỏa mãn $|A| = k$. Ta chứng minh $Q(k+1)$ đúng, nghĩa là $|\mathcal{P}(B)| = 2^{k+1}$ với mọi tập hợp B thỏa mãn $|B| = k+1$. Thật vậy, giả sử B là một tập hợp có $k+1$ phần tử. Gọi x là phần tử bất kỳ của B và đặt $C = B - \{x\}$. Chú ý rằng nếu D là một tập con của B thì hoặc D cũng là một tập con của C nếu $x \notin D$, hoặc D là hợp của một tập con của $C \cup \{x\}$ nếu $x \in D$. Do đó $|\mathcal{P}(B)| = 2|\mathcal{P}(C)|$, và theo giả thiết quy nạp $|\mathcal{P}(C)| = 2^k$. Suy ra $|\mathcal{P}(B)| = 2 \cdot 2^k = 2^{k+1}$

Hãy sử dụng phương pháp quy nạp để giải các bài tập sau

Bài tập 2

*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 nhau*



Hình: *Phủ kín các bàn cờ, ví dụ như (b) hoặc (c), bằng các khối hình chữ **L** như ở (a)*

(Bạn có thể thử phủ kín bàn cờ 8×8 ở https://nstarr.people.amherst.edu/puzzle.html)

Hãy sử dụng phương pháp quy nạp để giải các bài tập sau

Bài tập 3

Chứng minh với mọi số nguyên $n \geq 0$,

$$\sum_{i=0}^{n} (-1)^i = \begin{cases} 1 & \text{nếu } n \text{ chẵn} \\ 0 & \text{nếu } n \text{ lẻ} \end{cases}$$

Bài tập 4

Chứng minh rằng với mọi số nguyên $n \geq 4$, ta có $2^n \geq n^2$

Bài tập 5

*Chứng minh rằng với mọi số nguyên n "đủ lớn", ta có $2^n \geq n^3$ (**Gợi ý:** Với bài tập này, trước tiên bạn cần tìm được một số nguyên b và chứng minh bất đẳng thức đã cho đúng với mọi $n \geq b$)*

Bài tập 6

Chứng minh rằng với mọi số nguyên $n \geq 2$, các tập $A_1, A_2, \dots, A_n$ thỏa mãn $\overline{A_1 \cap A_2 \cap \cdots \cap A_n} = \overline{A_1} \cup \overline{A_2} \cup \cdots \cup \overline{A_n}$

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 $(P(1) \wedge P(2) \wedge \cdots \wedge P(k)) \to P(k+1)$ 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 lôgic,

$$(P(1) \wedge \forall k \in \mathbb{Z}^+ \; (\bigwedge_{j=1}^{k} P(j) \to P(k+1))) \to \forall n \in \mathbb{Z}^+ \; P(n),$$

trong đó $\bigwedge_{j=1}^{k} P(j) = P(1) \wedge P(2) \wedge \cdots \wedge P(k)$

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 $(P(b) \wedge P(b+1) \wedge \cdots \wedge P(k)) \to P(k+1)$ đúng với mọi số nguyên $k \geq b$

Theo ngôn ngữ lôgic,

$$(P(b) \wedge \forall k \in \mathbb{Z}^{\geq b}\, (\bigwedge_{j=b}^{k} P(j) \to P(k+1))) \to \forall n \in \mathbb{Z}^{\geq b}\, P(n)$$

Bài tập 7

Tương tự như Bài tập 1, hãy chứng minh nguyên lý quy nạp mạnh trên là đúng

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 $P(b)$, $P(b+1), \ldots, P(b+j)$ đúng, với j là một số nguyên dương cố định nào đó
- *Bước quy nạp (inductive step):* Chứng minh mệnh đề $(P(b) \wedge P(b+1) \wedge \cdots \wedge P(k)) \to P(k+1)$ đúng với mọi số nguyên $k \geq b+j$

Theo ngôn ngữ lôgic,

$$\left(\bigwedge_{i=0}^{j} P(b+i) \wedge \forall k \in \mathbb{Z}^{\geq b+j} \left(\bigwedge_{i=b}^{k} P(i) \to P(k+1)\right)\right) \to \forall n \in \mathbb{Z}^{\geq b}\, P(n)$$

- 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 $P(1), P(2), \ldots, P(k)$ 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, *nên dùng quy nạp yếu*
- Ngược lại, nếu bạn có thể chứng minh $P(k+1)$ từ giả thiết $P(j)$ đúng với mọi $j \le k$, nhưng không rõ làm sao để trực tiếp chứng minh $P(k) \to P(k+1)$, *nên dùng quy nạp mạnh*

Ví dụ 4

Cho $P(n)$ là vị từ xác định trên miền $\mathbb{Z}^{\geq 2}$ như sau

n có thể được biểu diễn dưới dạng tích của các số nguyên tố

Ta chứng minh $\forall n \in \mathbb{Z}^{\geq 2}\ P(n)$ bằng phương pháp 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 $2 \leq j \leq k$ với $k \geq 2$ 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

Ví dụ 5

Cho vị từ $P(n)$ với $n \in \mathbb{Z}^{\geq 12}$

$$n = 4a + 5b \text{ với } a, b \in \mathbb{Z}$$

Ta chứng minh $\forall n \in \mathbb{Z}^{\geq 12}\ P(n)$ bằng phương pháp quy nạp mạnh

- **Bước cơ sở.** Ta chứng minh $P(12), P(13), P(14), P(15)$ đều đúng. Thật vậy, $12 = 4 \cdot 3$, $13 = 4 \cdot 2 + 5 \cdot 1$, $14 = 4 \cdot 1 + 5 \cdot 2$, và $15 = 5 \cdot 3$
- **Bước quy nạp.** Giả sử với mọi số nguyên cố định $k \geq 15$ bất kỳ, $P(m)$ đúng với mọi số nguyên m thỏa mãn $12 \leq m \leq k$. 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 8

Chứng minh ví dụ trên bằng quy nạp yếu

Bài tập 9

Gọi $P(n)$ là vị từ $n = 3a + 5b$ với $a, b \in \mathbb{Z}$. Bài tập này mô tả cách chứng minh $P(n)$ đúng với mọi số nguyên $n \geq 8$ bằng phương pháp quy nạp mạnh

(a) *Để hoàn thành bước cơ sở, hãy chứng minh rằng các mệnh đề $P(8)$, $P(9)$, và $P(10)$ là đúng*

(b) *Giả thiết quy nạp là gì?*

(c) *Trong bước quy nạp, bạn cần chứng minh điều gì?*

(d) *Hãy hoàn thành bước quy nạp cho $n \geq 10$*

(e) *Giải thích tại sao những bước trên cho thấy rằng $P(n)$ đúng với mọi $n \geq 8$*

Bài tập 10

Chỉ sử dụng các tờ tiền mệnh giá 20000 VND và 50000 VND, các bạn có thể tạo thành cọc tiền có những giá trị như thế nào? Hãy chứng minh câu trả lời của bạn bằng phương pháp quy nạp mạnh

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

- **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

$$\begin{aligned}
\sum_{i=1}^{k+1} i &= \left(\sum_{i=1}^{k} i\right) + (k+1) \\
&= \frac{(k+\frac{1}{2})^2}{2} + (k+1) \qquad \text{giả thiết quy nạp} \\
&= \frac{(k+\frac{1}{2})^2 + 2(k+\frac{1}{2}) + 1}{2} \\
&= \frac{(k+1+\frac{1}{2})^2}{2}
\end{aligned}$$

Câu hỏi

Chứng minh này sai ở đâu?

Ví dụ 7 ([Gunderson and Rosen 2010])

Để chứng minh *mọi số nguyên dương đều bằng nhau*, bước đầu tiên ta xét

$$P(n) := \text{“}\textit{Nếu } n = \max\{a, b\} \textit{ với } a, b \in \mathbb{Z}^+ \textit{ thì } a = b\text{”}$$

Nếu $P(n)$ đúng với mọi $n \in \mathbb{Z}^+$, thì với mọi $x, y \in \mathbb{Z}^+$, ta chọn $n = \max\{x, y\}$ và theo $P(n)$ ta có $x = y$. Do đó mọi số nguyên dương đều bằng nhau. Ta chứng minh $P(n)$ đúng với mọi $n \in \mathbb{Z}^+$ bằng phương pháp quy nạp

- **Bước cơ sở.** Với $n = 1$, ta có $\max\{a, b\} = 1$ và $a, b \in \mathbb{Z}^+$, 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. Ta chứng minh $P(k + 1)$ đúng

Câu hỏi

Chứng minh này sai ở đâu?

Thật vậy, giả sử hai số $c, d \in \mathbb{Z}^+$ thỏa mãn $\max\{c, d\} = k + 1$. Do đó, $\max\{c - 1, d - 1\} = k$. Theo giả thiết quy nạp, $c - 1 = d - 1$, và do đó $c = d$

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

- **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

Câu hỏi

Chứng minh này sai ở đâu?

$$H_1 \quad H_2 \quad H_3 \quad H_4$$

- 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

- *Hàm định nghĩa bằng đệ quy (recursive function)*
  $f : \mathbb{N} \to A$ 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 $n > b$

Ví dụ 9

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:** $g_n = ng_{n-1}$ $(n \geq 1)$

Ví dụ 10 (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 mọi i thỏa mãn $1 \leq i \leq k$, ta có $f_i < 2^i$. Ta chứng minh $f_{k+1} < 2^{k+1}$. Thật vậy,

$$\begin{aligned} f_{k+1} &= f_k + f_{k-1} \\ &< 2^k + 2^{k-1} && \text{giả thiết quy nạp} \\ &< 2^k + 2^k = 2^{k+1} \end{aligned}$$

Ví dụ 11 (Chứng minh tính chất của hàm sử dụng định nghĩa đệ quy)

Cho dãy Fibonacci $\{f_n\}$. Ta chứng minh $f_n > \alpha^{n-2}$ *với mọi số tự nhiên* $n \geq 3$ *và* $\alpha = \dfrac{1+\sqrt{5}}{2} \approx 1.61803$ bằng quy nạp mạnh

- **Bước cơ sở:** Với $n = 3$, ta có $f_3 = 2 > \alpha^{3-2} = \alpha$. Với $n = 4$, ta có $f_4 = 3 > \alpha^{4-2} = \alpha^2 = \left(\dfrac{1+\sqrt{5}}{2}\right)^2 = \dfrac{3+\sqrt{5}}{2} = \alpha + 1 \approx 2.61803$
- **Bước quy nạp:** Giả sử với mọi i thỏa mãn $3 \leq i \leq k$ ta có $f_i > \alpha^{i-2}$. Ta chứng minh $f_{k+1} > \alpha^{k-1}$. Thật vậy,

$$\begin{aligned} f_{k+1} &= f_k + f_{k-1} \\ &> \alpha^{k-2} + \alpha^{k-3} && \text{giả thiết quy nạp} \\ &= \alpha^{k-3}(\alpha+1) \\ &= \alpha^{k-3}\alpha^2 = \alpha^{k-1} && \text{do } \alpha^2 = \alpha + 1 \end{aligned}$$

- *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.

Ví dụ 12

- 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

Ví dụ 13

- Tập *các công thức được tạo đúng quy tắc (well-formed formulae)* trong lôgic 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 $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \to B)$, và $(A \leftrightarrow B)$ cũng thế
  - Ví dụ, với các mệnh đề nguyên tử p, q, $((p \vee q) \to (q \wedge \mathbf{F}))$ là công thức được tạo đúng quy tắc, còn $\wedge pq$, $pq\wedge$, và $p \wedge q$ thì không phải (Tại sao?)

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

Ví dụ 14

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

Ví dụ 15

Ta chứng minh bằng quy nạp theo cấu trúc rằng *mọi công thức được tạo đúng quy tắc trong lôgic mệnh đề có số dấu ngoặc đơn trái "(" bằng số dấu ngoặc đơn phải ")"*

- **Bước cơ sở:** Các mệnh đề $\mathbf{T}$, $\mathbf{F}$, 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

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.

Part I

Phụ lục

Nội dung

Chứng minh (Quy nạp mạnh là đúng).

- Giả sử $P(1)$ đúng và với mọi $k \in \mathbb{Z}^+$, $(P(1) \wedge P(2) \wedge \cdots \wedge P(k)) \to P(k+1)$ 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 $S = \{n \mid n \in \mathbb{Z}^+$ và $P(n)$ 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 $P(1) \wedge P(2) \wedge \cdots \wedge P(m-1)$ đúng
- Do $(P(1) \wedge P(2) \wedge \cdots \wedge P(k)) \to P(k+1)$ đúng với mọi $k \in \mathbb{Z}^+$, ta có $(P(1) \wedge P(2) \wedge \cdots \wedge P(m-1)) \to P(m)$ đúng
- Kết hợp với $P(1) \wedge P(2) \wedge \cdots \wedge P(m-1)$ 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 yếu $\Rightarrow$ Quy nạp mạnh.

(1) Giả sử với mọi $R(n)$, ta có $(R(1) \land \forall k \in \mathbb{Z}^+ (R(k) \to R(k+1))) \to \forall n \in \mathbb{Z}^+ R(n)$ đúng, nghĩa là, quy nạp yếu đúng

(2) Giả sử với vị từ $P(n)$ ta có $P(1) \land \forall k \in \mathbb{Z}^+ \bigwedge_{j=1}^{k} P(j) \to P(k+1)$ 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 $Q(n) = \bigwedge_{j=1}^{n} P(n)$ với mọi $n \in \mathbb{Z}^+$

(4) Từ (1), ta có $(Q(1) \land \forall k \in \mathbb{Z}^+ (Q(k) \to Q(k+1))) \to \forall n \in \mathbb{Z}^+ Q(n)$ 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 $p \to q \equiv p \to p \land q$ (Tại sao?).

(7) Do đó $Q(k) \to Q(k+1) \equiv \bigwedge_{j=1}^{k} P(k) \to \bigwedge_{j=1}^{k+1} P(k+1) \equiv \bigwedge_{j=1}^{k} P(k) \to \bigwedge_{j=1}^{k} P(k) \land P(k+1) \equiv \bigwedge_{j=1}^{k} P(k) \to P(k+1)$

(8) Thay (5) và (7) vào (4), ta có điều phải chứng minh

$\square$

Quy nạp mạnh $\Rightarrow$ Quy nạp yếu.

(1) Giả sử với mọi $R(n)$ ta có $(R(1) \land \forall k \in \mathbb{Z}^+ \, (\bigwedge_{j=1}^{k} R(j) \to R(k+1))) \to \forall n \in \mathbb{Z}^+ \, R(n)$ đúng, nghĩa là, quy nạp mạnh đúng

(2) Giả sử với vị từ $P(n)$ ta có $P(1) \land \forall k \in \mathbb{Z}^+ \, (P(k) \to P(k+1))$ 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 $p \to q$ đúng thì $p \land r \to q$ cũng đúng (Tại sao?)

(4) Áp dụng (1) với $P(n)$, ta có $(P(1) \land \forall k \in \mathbb{Z}^+ \, (\bigwedge_{j=1}^{k} P(j) \to P(k+1))) \to \forall n \in \mathbb{Z}^+ \, P(n)$ đúng.

(5) Từ (2), ta có $P(1)$ đúng và $\forall k \in \mathbb{Z}^+ \, (P(k) \to P(k+1))$ đúng. Kết hợp với (3), ta có $\forall k \in \mathbb{Z}^+ \, (\bigwedge_{j=1}^{k} P(j) \to P(k+1))$ đúng

(6) Từ (4) và (5) ta có điều phải chứng minh

$\square$

VNU-HUS MAT3500: Toán rời rạc

Thuật toán I

Giới thiệu, một số thuật toán tìm kiếm và sắp xếp, độ tăng của hàm

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
- Định nghĩa và một số khái niệm
- Một số quy tắc viết mã giả
- Bất biến vòng lặp

Tìm kiếm và Sắp xếp
- Một số thuật toán tìm kiếm
- Một số thuật toán sắp xếp

Độ tăng của các hàm
- Giới thiệu
- Một số ký hiệu quan trọng

Thuật toán

Định nghĩa và một số khái niệm

- 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*

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

Thuật toán

Định nghĩa và một số khái niệm

- Một *thuật toán* cũng 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*, *giả mã (pseudocode)*, hoặc *sơ đồ khối (flowchart)*
- 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
    - 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, thuật toán cần cho kết quả đầu ra đúng

  (3) Một phân tích về *thời gian chạy (running time)* của thuật toán

Sau đây, ta sẽ trình bày

- Ví dụ về một số *cách mô tả một thuật toán*
  - Ngôn ngữ thông thường
  - Sơ đồ khối
  - Mã giả
- Ví dụ về một *phương pháp chứng minh tính đúng đắn của một thuật toán*: phương pháp sử dụng *bất biến vòng lặp (loop invariant)*. Ta sẽ minh họa phương pháp này bằng cách giới thiệu ý tưởng chứng minh tính đúng đắn của các thuật toán sẽ được mô tả

Ví dụ 1 (Mô tả thuật toán bằng ngôn ngữ thông thường)

- **Bài toán:**
  - **Input:** $a_1, a_2, \ldots, a_n$: dãy số nguyên
  - **Output:** Giá trị của phần tử lớn nhất trong dãy
- **Tìm giá trị của phần tử lớn nhất:**
  (1) Gán giá trị của một biến tạm thời v (phần tử lớn nhất đến thời điểm hiện tại) bằng a_1
  (2) Xét phần tử ngay tiếp theo trong dãy
  (3) Nếu phần tử đó lớn hơn v thì gán giá trị của v bằng giá trị của phần tử
  (4) Lặp lại (2) và (3) cho đến khi không còn phần tử nào để xét
  (5) Trả lại giá trị của v

Thuật toán I

Hoàng Anh Đức

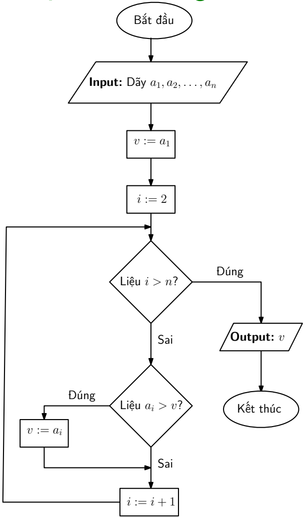
Thuật toán
7 Định nghĩa và một số khái niệm
Một số quy tắc viết mã giả
Bất biến vòng lặp
Tìm kiếm và Sắp xếp
Một số thuật toán tìm kiếm
Một số thuật toán sắp xếp
Độ tăng của các hàm
Giới thiệu
Một số ký hiệu quan trọng

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
$\boxed{7}$	$\boxed{12}$	$\boxed{5}$	$\boxed{16}$	$\boxed{9}$
	$i = 2$	$i = 3$	$i = 4$	$i = 5$
$v = 7$	$v = 12$	$v = 12$	$v = 16$	$v = 16$

Ví dụ 3 (Mô tả thuật toán bằng sơ đồ khối)



Ví dụ 4 (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

$v := a_1$ `// phần tử lớn nhất đến hiện tại`
for $i := 2$ **to** n **do** `// lần lượt xét` $a_2, \dots, a_n$
  if $a_i > v$ **then** `//` a_i `> phần tử lớn nhất hiện tại?`
    $v := a_i$ `// bây giờ` v `lớn nhất trong` $a_1, \dots, a_i$
return v

Thuật toán

Một số quy tắc viết giả mã

- *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
- Một *phép gán (assignment)* thường có dạng *variable* $:=$ *expression*, trong đó biểu thức expression ở vế phải sẽ được tính toán và kết quả tính toán được lưu trữ ở vị trí trong bộ nhớ tương ứng với biến variable
- Trong *cấu trúc điều kiện (conditional statement)* ***if*** *condition* ***then*** S_1 ***else*** S_2, biểu thức condition được tính toán và sẽ cho ra giá trị cuối cùng là `True` (đúng) hoặc `False` (sai). Nếu `True` thì đoạn mã S_1 sẽ được thực hiện, còn ngược lại thì S_2 sẽ được thực hiện. Sau khi S_1 hoặc S_2 được thực hiện, các lệnh ngay tiếp sau cấu trúc điều kiện sẽ được thực hiện

Thuật toán

Một số quy tắc viết giả mã

- Trong *vòng lặp **for** (**for** loop)* ***for*** $i = 1$ ***to*** n ***do*** S, giá trị ban đầu của biến i được gán bằng 1. Nếu giá trị của i nhỏ hơn hoặc bằng n, đoạn mã S sẽ được thực hiện. Đoạn mã S có thể sử dụng biến i hoặc không. Sau mỗi lần thực hiện S (*lần lặp (iteration)*), biến i được tăng thêm 1 và được kiểm tra xem giá trị sau khi tăng của i có phải vẫn nhỏ hơn hoặc bằng n hay không. Nếu kết quả là `True` thì vòng lặp được thực hiện thêm một lần nữa với giá trị mới của i. Ngược lại, các lệnh tiếp theo ngay sau vòng **for** được thực hiện
- Trong *vòng lặp **while** (**while** loop)* ***while*** *condition* ***do*** S, đoạn mã S được thực hiện bất kể khi nào giá trị của biểu thức condition còn đúng (`True`). Nếu condition sai ngay khi bắt đầu vòng lặp **while** thì S không bao giờ được thực hiện
- Mọi thứ nằm sau các dấu `//` hoặc nằm giữa `/*` và `*/` là các nhận xét hoặc chú thích mà chương trình sẽ bỏ qua

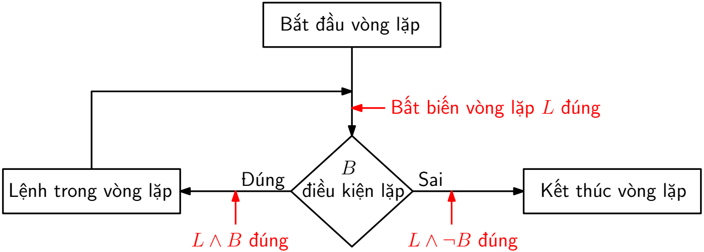
- Có rất nhiều phương pháp khác nhau để chứng minh tính đúng đắn của một thuật toán
- Một trong số đó là sử dụng *bất biến vòng lặp (loop invariant)*—một phương pháp được xây dựng dựa trên phương pháp quy nạp toán học
  - *Vòng lặp (loop):* **for**, **while**, v.v...
  - Một *bất biến vòng lặp* là *một phát biểu luôn đúng trước và sau mỗi lần lặp (iteration) của một vòng lặp (loop)*

Ta cần chỉ ra ba điều về một bất biến vòng lặp L

Khởi động (Initialization) L đúng trước lần lặp đầu tiên của vòng lặp

Duy trì (Maintainance) 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



Hình: 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 $a_1, \dots, a_n$

$$L := \textit{Ở trước lần lặp với biến } i,\ v = \max\{a_1, a_2, \dots, a_{i-1}\}$$

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

Thuật toán

Một số thuật toán tìm kiếm

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)

- **Bài toán:**
  - **Input:** $a_1, \ldots, a_n$: 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:** Lần lượt xét các phần tử trong dãy cho đến khi tìm được x hoặc không còn phần tử nào để xét

Ví dụ 6 (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
$\boxed{2}$	$\boxed{5}$	$\boxed{6}$	$\boxed{8}$	12
$i = 1$	$i = 2$	$i = 3$	$i = 4$	
$\neq x$	$\neq x$	$\neq x$	$= x$	

Thuật toán
Tìm kiếm tuyến tính

Thuật toán 2: Tìm kiếm tuyến tính (Linear Search)

Input: $a_1, \ldots, a_n$: dãy số nguyên, x: số nguyên
Output: Chỉ số i thỏa mãn $x = a_i$ hoặc 0 nếu x không có trong dãy

```
i := 1                                  // Bắt đầu từ đầu dãy
while i ≤ n và x ≠ a_i do               // Chưa xong và chưa tìm thấy
    i := i + 1                          // Đi tới vị trí tiếp theo trong dãy
if i ≤ n then
    location := i                       // Tìm thấy x trong dãy
else
    location := 0                       // Không tìm thấy x trong dãy
return location
```

Một bất biến vòng lặp trong Thuật toán 2 (vòng **while** ở Dòng 2–3) tìm kiếm tuyến tính số nguyên x trong dãy $a_1, \dots, a_n$

$$L := \textit{Ở trước lần lặp với biến } i, \; x \notin \{a_1, \dots, a_{i-1}\}$$

- **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

- **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 7 (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
			$i = j$	
			$= x$	

Thuật toán
Tìm kiếm nhị phân

Thuật toán 3: 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

$i := 1$ `// Chỉ số bắt đầu khoảng tìm kiếm`
$j := n$ `// Chỉ số kết thúc khoảng tìm kiếm`
while $i < j$ **do** `// Khi khoảng tìm kiếm có` > 1 `phần tử`
  $m := \lfloor (i+j)/2 \rfloor$ `// Chỉ số của phần tử ở giữa`
  if $x > a_m$ **then**
    $i := m + 1$
  else
    $j := m$
if $x = a_i$ **then**
  location $:= i$
else
  location $:= 0$
return *location*

Một bất biến vòng lặp trong Thuật toán 3 (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 3 trả lại vị trí chính xác của x hoặc kết luận không tìm được x (Dòng 9–13)

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)

- **Bài toán:**
  - **Input:** $a_1, a_2, \ldots, a_n$: 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), $\ldots$, (a_{n-1}, a_n). Lúc này, a_n là phần tử lớn nhất trong dãy
  (4) Lặp lại (1) – (3) với dãy $a_1, \ldots, a_{n-1}$, và sau đó với dãy $a_1, \ldots, a_{n-2}$, dãy $a_1, \ldots, a_{n-3}$, $\ldots$, cho đến dãy a_1, a_2

Ví dụ 8

- **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

Thuật toán 4: Sắp xếp nổi bọt (Bubble Sort)

Input: $a_1, a_2, \ldots, a_n$: dãy số thực $(n \geq 2)$
Output: Dãy đã cho được sắp xếp theo thứ tự tăng dần

for $i := 1$ **to** $n - 1$ **do** `// Lặp lại` $n-1$ `lần`
    for $j := 1$ **to** $n - i$ **do**
        if $a_j > a_{j+1}$ **then**
            Hoán đổi giá trị của a_j và a_{j+1}
    `//` $a_{n-i+1}, \ldots, a_n$ `đã được sắp xếp`
`//` $a_1, \ldots, a_n$ `đã được sắp xếp`

Thuật toán 4 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 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) Xét a_2. Tìm vị trí trong dãy 1 phần tử a_1 để chèn a_2 bằng cách duyệt toàn bộ các phần tử trong dãy từ phải sang trái và đẩy mọi phần tử lớn hơn a_2 sang phải một bước. Chèn a_2 vào ngay sau phần tử đầu tiên nhỏ hơn hoặc bằng a_2 khi duyệt dãy ở trên. Đến đây dãy a_1, a_2 đã được sắp xếp theo đúng thứ tự
  (2) Xét a_3. Tìm vị trí trong dãy 2 phần tử a_1, a_2 để chèn a_3 bằng cách duyệt toàn bộ các phần tử trong dãy từ phải sang trái và đẩy mọi phần tử lớn hơn a_3 sang phải một bước. Chèn a_3 vào ngay sau phần tử đầu tiên nhỏ hơn hoặc bằng a_3 khi duyệt dãy ở trên. Đến đây dãy a_1, a_2, a_3 đã được sắp xếp theo đúng thứ tự
  (3) Tiếp tục với $a_4, a_5, \dots, a_n$. Cuối cùng ta thu được dãy $a_1, \dots, a_n$ đã được sắp xếp theo đúng thứ tự

- **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

Thuật toán 5: Sắp xếp chèn (Insertion Sort)

Input: $a_1, a_2, \ldots, a_n$: dãy số thực $(n \geq 2)$
Output: Dãy đã cho được sắp xếp theo thứ tự tăng dần

```
for i = 2 to n do
    m := a_i        // m sắp được chèn vào dãy a_1, ..., a_{i-1}
    j := i - 1
    while j ≥ 1 và m < a_j do        // Nếu m < a_j, đẩy a_j
      sang phải để có chỗ chèn m
        a_{j+1} := a_j
        j := j - 1
    a_{j+1} := m                     // Chèn m
    // Dãy a_1, ..., a_i đã được sắp thứ tự
```

Thuật toán 5 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, $m \leq \min\{a_{j+1}, \dots, a_i\}$

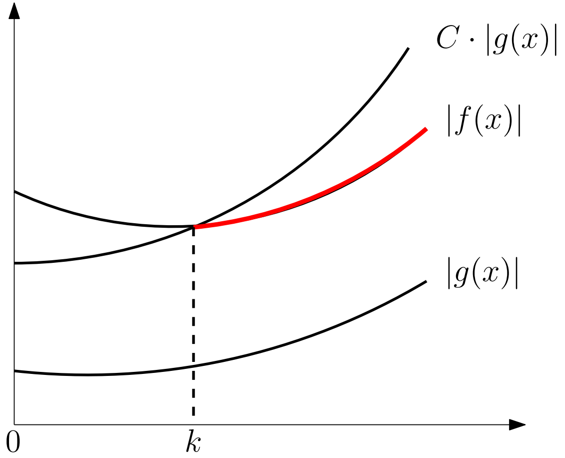
- Việc *phân tích (analysis) một thuật toán* yêu cầu cung cấp một xấp xỉ về *thời gian (time)* và *bộ nhớ (space)* cần thiết để thực hiện một thuật toán
- *Độ phức tạp (complexity)* của một thuật toán là lượng thời gian và bộ nhớ cần để thực hiện một thuật toán
  - Thường được thể hiện thông qua các hàm của *kích thước của đầu vào (input size)*
- Để đánh giá và so sánh độ phức tạp của các thuật toán khác nhau, ta giới thiệu *ký hiệu O-lớn (big-O notation)* và mô tả cách xấp xỉ độ tăng của các hàm thông qua ký hiệu này và từ đó xấp xỉ độ phức tạp của các thuật toán
- Với các hàm $f : \mathbb{R} \to \mathbb{R}$ hoặc $f : \mathbb{N} \to \mathbb{R}$, trong nhiều trường hợp ta cần tìm hiểu xem chúng tăng nhanh đến mức nào
  - So sánh các hàm: Nếu f *tăng nhanh hơn* g thì $f(x) \geq g(x)$ với "giá trị x đủ lớn"
  - So sánh tính hiệu quả của các thuật toán khác nhau cùng giải quyết một bài toán

Ký hiệu O-lớn

Cho f và g là các hàm $\mathbb{R} \to \mathbb{R}$. 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 $x > k$

- $\exists C, k\, \forall x > k\, (|f(x)| \leq C|g(x)|)$
- f là $O(g)$ nếu từ sau điểm k nào đó, giá trị của hàm f không vượt quá giá trị của một hằng số nhân với giá trị của hàm 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 liệu f có là $O(g)$ hay không, chỉ cần một cặp bằng chứng là đủ

Độ tăng của các hàm
Ký hiệu O-lớn



f là $O(g)$

Ví dụ 9

Ta chứng minh hàm f 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:

- Chú ý rằng khi $x > 2$, ta có $2x \leq x^2$ và $1 \leq x^2$
- Do đó với mọi $x > 1$, ta có

$$f(x) = |x^2 + 2x + 1| \leq |x^2 + x^2 + x^2| = 3|x^2|$$

- Ta chọn $C = 3$ và $k = 2$

Bài tập 1

Chứng minh

(a) $7x$ *là* $O(x^3)$

(b) x^3 *không là* $O(x^2)$

(c) $1 + 2 + \cdots + n$ *là* $O(n^2)$

(d) $n! = 1 \times 2 \times \cdots \times n$ *là* $O(n^n)$

(e) $\log(n!)$ *là* $O(n \log n)$ *(Nếu không đề cập gì thêm thì* $\log n = \log_2 n$*)*

(f) n^2 *là* $O(2^n)$

(g) $\log n$ *là* $O(n)$

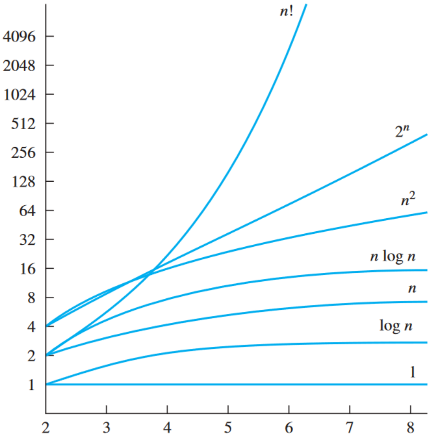
(h) *Với các hằng số* $b > 1$ *và* $k > 0$, $\log_b(n^k)$ *là* $O(\log n)$

Bài tập 2

Chứng minh rằng nếu $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ *với* $a_0, a_1, \ldots, a_n$ *là các số thực (nghĩa là,* $f(x)$ *là một đa thức bậc* n*) thì* f *là* $O(x^n)$

Độ tăng của các hàm

Ký hiệu O-lớn



Hình: Độ tăng của một số hàm thường dùng khi đánh giá với ký hiệu O-lớn [Rosen 2012]

Một số xấp xỉ hữu ích

- Nếu $d > c > 1$, thì n^c là $O(n^d)$, nhưng n^d không là $O(n^c)$
- Nếu $b > 1$ và c, d là các số dương, thì $(\log_b n)^c$ là $O(n^d)$ nhưng n^d không là $O((\log_b n)^c)$
- Nếu $b > 1$ và d là số dương, thì n^d là $O(b^n)$ nhưng b^n không là $O(n^d)$
- Nếu $c > b > 1$, thì b^n là $O(c^n)$ nhưng c^n không là $O(b^n)$
- Nếu $f_1(x)$ là $O(g_1(x))$ và $f_2(x)$ là $O(g_2(x))$ thì $(f_1 + f_2)(x)$ là $O(\max(|g_1(x)|, |g_2(x)|))$
- Nếu f_1 là $O(g_1)$ và f_2 là $O(g_2)$ thì $f_1 \circ f_2$ là $O(g_1 \circ g_2)$

Độ tăng của các hàm

Ký hiệu Ω-lớn

Ký hiệu Ω-lớn

Cho f và g là các hàm $\mathbb{R} \to \mathbb{R}$. 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 $x > k$

Bài tập 3

Chứng minh rằng f là $\Omega(g)$ khi và chỉ khi g là $O(f)$

Ký hiệu Θ-lớn

Cho f và g là các hàm $\mathbb{R} \to \mathbb{R}$. Ta nói rằng *$f$ là $\Theta(g)$* nếu f là $O(g)$ và f là $\Omega(g)$

Bài tập 4

Chứng minh rằng $1 + 2 + \cdots + n$ *là* $\Theta(n^2)$

Với các hàm f và g từ $\mathbb{R}$ đến $\mathbb{R}$

- f là $O(g)$ tương tự "$\leq$"

$$\exists C, k \ \forall x > k \quad |f(x)| \leq C|g(x)|$$

- f là $\Omega(g)$ tương tự "$\geq$"

$$\exists C > 0, k \ \forall x > k \quad |f(x)| \geq C|g(x)|$$

- f là $\Theta(g)$ tương tự "$=$"

$$\exists C_1, C_2, k \ \forall x > k \quad C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$$

VNU-HUS MAT3500: Toán rời rạc

Thuật toán II

Độ phức tạp tính toán, thuật toán tham lam, thuật toán 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

Độ phức tạp tính toán
Giới thiệu

- Một thuật toán cần *luôn xuất ra kết quả đúng* và tốt nhất là *có hiệu suất cao*
- Chúng ta đã tìm hiểu một số *ví dụ* về các thuật toán tìm kiếm và sắp xếp trong bài giảng trước và phương pháp *chứng minh tính đúng đắn* của chúng dựa trên bất biến vòng lặp (loop invariant).
- *Độ phức tạp (complexity)* của một tính toán là một cách đo độ "khó" của việc thực hiện tính toán đó
  - *Độ phức tạp theo thời gian (time complexity):* Số các toán tử hoặc số bước cần thiết
  - *Độ phức tạp theo bộ nhớ (space complexity):* Số các bit trong bộ nhớ cần thiết
- Phần lớn các thuật toán có độ phức tạp khác nhau đối với các đầu vào có kích thước khác nhau
  - Tìm kiếm trong một dãy dài thường tốn nhiều thời gian hơn tìm kiếm trong một dãy ngắn
- Do đó, độ phức tạp tính toán thường được biểu diễn dưới dạng một *hàm (function) của kích thước đầu vào (input size)*

- Khi xét độ phức tạp của một thuật toán, ta không quan tâm đến số lượng chính xác các toán tử/bit cần thiết mà chỉ cần một *đánh giá tiệm cận (asymptotic estimate)*
- Một công cụ hữu ích cho việc đánh giá độ phức tạp tính toán là các ký hiệu O-lớn, Ω-lớn, và Θ-lớn
- Chúng ta sẽ tập trung vào *độ phức tạp tính toán theo thời gian (time complexity)* (chủ yếu là *trong trường hợp xấu nhất*)
  - *Độ phức tạp trong trường hợp xấu nhất (worst-case complexity):* xấp xỉ thời gian nhiều nhất cần để giải quyết các trường hợp đầu vào với mỗi kích thước đầu vào
  - *Độ phức tạp trong trường hợp trung gian (average-case complexity):* xấp xỉ thời gian trung bình cần để giải quyết các trường hợp đầu vào với mỗi kích thước đầu vào
  - *Độ phức tạp trong trường hợp tốt nhất (best-case complexity):* xấp xỉ thời gian ít nhất cần để giải quyết các trường hợp đầu vào với mỗi kích thước đầu vào

Độ phức tạp tính toán
Giới thiệu

Với các hàm f và g từ $\mathbb{R}$ đến $\mathbb{R}$

- f là $O(g)$ tương tự "$\leq$"

$$\exists C, k\ \forall x > k \quad |f(x)| \leq C|g(x)|$$

- f là $\Omega(g)$ tương tự "$\geq$"

$$\exists C > 0, k\ \forall x > k \quad |f(x)| \geq C|g(x)|$$

- f là $\Theta(g)$ tương tự "$=$"

$$\exists C_1, C_2, k\ \forall x > k \quad C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$$

Độ phức tạp tính toán
Chú ý

- 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 nhân giá trị dương.) Định nghĩa theo cách này không hoàn toàn chặt chẽ. Ví dụ như hàm $\log n$ có thể nhận giá trị âm với n nhỏ

- *Để đơn giản, ta thường mô tả thời gian chạy của thuật toán theo O-lớn*. Chú ý rằng với nhiều thuật toán, ta có thể thu được đánh giá tốt hơn với xấp xỉ theo Θ-lớn. Tuy nhiên không phải lúc nào ta cũng làm được điều này
- Nhiều tác giả viết "$f(x)$ là $O(g(x))$" trong khi điều họ thực sự muốn thể hiện là "$f(x)$ là $\Theta(g(x))$"

Một số quy tắc tính độ phức tạp thuật toán theo thời gian

- 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 $O(1)$
  - Chú ý rằng với phần cứng hữu hạn (64-bit CPU chỉ biểu diễn được tối đa các số nguyên có giá trị nhỏ hơn 2^{64} dưới dạng số nhị phân), việc cộng hai số nguyên độ dài n bit biểu diễn dưới dạng số nhị phân có độ phức tạp $O(n)$
- Độ phức tạp của một chuỗi tuần tự các bước là tổng của độ phức tạp của từng bước
- Thời gian thực hiện cấu trúc **if...then...else** là thời gian lớn nhất thực hiện các lệnh sau **then** hoặc sau **else** và thời gian kiểm tra điều kiện
- 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

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)

Độ phức tạp tính toán

Thuật toán tìm giá trị lớn nhất

Thuật toán 1: Tìm giá trị của phần tử lớn nhất

Input: $a_1, a_2, \ldots, a_n$: dãy số nguyên

Output: Giá trị của phần tử lớn nhất trong dãy

```
v := a_1                         t_1
for i := 2 to n do               t_2
    if a_i > v then              t_4^i
        v := a_i                 t_5
return v                         t_3
```

$T(n) = t_1 + t_2 + t_3$

t_1, t_3, t_5 là $O(1)$

$\boxed{T(n) \text{ là } O(n)}$ (xấu nhất)

$\boxed{T(n) \text{ là } O(n)}$ (tốt nhất)

$$t_2 = \sum_{i=2}^{n} \left[t_4^i + (\text{thời gian kiểm tra } i \le n) \right] = \sum_{i=2}^{n} \left[t_4^i + O(1) \right]$$

$$t_4^i = t_5 + (\text{thời gian kiểm tra } a_i > v) = O(1)$$

Độ phức tạp tính toán
Tìm kiếm tuyến tính

Thuật toán 2: 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

$i := 1$ — t_1
while $i \leq n$ và $x \neq a_i$ **do** — t_2
  $i := i + 1$ — t_5^i
if $i \leq n$ **then** — t_3
  location $:= i$ — t_6
else
  location $:= 0$ — t_7
return *location* — t_4

$T(n) = t_1 + t_2 + t_3 + t_4$

$t_1, t_4, t_5^i, t_6, t_7$ là $O(1)$

$T(n)$ là $O(n)$ (xấu nhất)

$T(n)$ là $O(1)$ (tốt nhất)

$$t_2 = \sum_{\{i \mid i \leq n \wedge x \neq a_i\}} \left[t_5^i + (\text{thời gian kiểm tra } i \leq n \text{ và } x \neq a_i) \right]$$

$$t_3 = \max\{t_6, t_7\} + (\text{thời gian kiểm tra } i \leq n)$$

Thuật toán 3: 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

$i := 1$
$j := n$ — t_1
while $i < j$ **do**
    $m := \lfloor (i+j)/2 \rfloor$ — $t_5^{i,j}$
    if $x > a_m$ **then**
        $i := m + 1$ — t_7
    else — $t_6^{i,j}$
        $j := m$ — t_8
— t_2
if $x = a_i$ **then**
    location $:= i$ — t_9
else — t_3
    location $:= 0$ — t_{10}
return *location* — t_4

$$T(n) = t_1 + t_2 + t_3 + t_4$$

$t_1, t_4, t_5^{i,j}, t_7, \dots, t_{10}$ là $O(1)$

$$t_2 = \sum_{\text{cặp } i,j \text{ mỗi lần lặp } \textbf{while}} \Big[(t_5^{i,j} + t_6^{i,j}) + \text{(thời gian kiểm tra } i < j)\Big]$$

$$t_6^{i,j} = \max\{t_7, t_8\} + \text{(thời gian kiểm tra } x > a_m)$$

$$t_3 = \max\{t_9, t_{10}\} + \text{(thời gian kiểm tra } x = a_i)$$

cặp i, j mỗi lần lặp **while** là $O(\log n)$ (ví dụ $n = 2^k$)

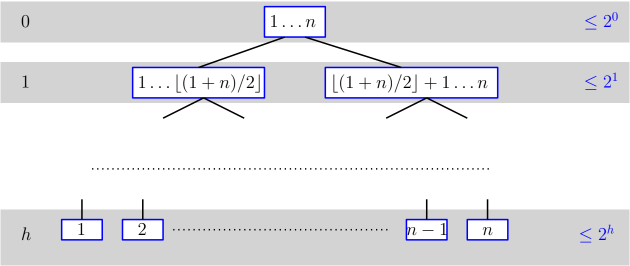
$T(n)$ là $O(\log n)$ (xấu nhất)

$T(n)$ là $O(\log n)$ (tốt nhất)

Độ phức tạp tính toán
Tìm kiếm nhị phân

cặp i, j mỗi lần lặp **while** là $O(\log n)$

- Đánh số các lần lặp **while** lần lượt bằng $0, 1, 2, \dots, h$
- Mỗi hàng $0, 1, 2, \dots, h$ liệt kê các khoảng $i \dots j$ có khả năng xuất hiện ở lần lặp đánh số tương ứng.
- Vòng lặp **while** dừng ở lần lặp h khi mọi khoảng $i \dots j$ chỉ có một phần tử $i = j$
- Ở mỗi hàng $k \in \{1, \dots, h\}$, tổng số phần tử của tất cả các khoảng là $\leq n$, và có $\leq 2^k$ khoảng (Hình dưới mô tả trường hợp $n = 2^h$)



Thuật toán 4: Sắp xếp nổi bọt (Bubble Sort)

Input: $a_1, a_2, \ldots, a_n$: dãy số thực $(n \geq 2)$

Output: Dãy đã cho được sắp xếp theo thứ tự tăng dần

```
for i := 1 to n − 1 do
    for j := 1 to n − i do
        if a_j > a_{j+1} then
            Hoán đổi giá trị của a_j và a_{j+1}   — t_4
                                                   t_3^j   t_2^i   t_1
```

$$T(n) = t_1 = \sum_{i=1}^{n-1}\left[t_2^i + (\text{t.g. kiểm tra } i \leq n-1)\right] = \sum_{i=1}^{n-1}\left[t_2^i + O(1)\right]$$

$$t_2^i = \sum_{j=1}^{n-i}\left[t_3^j + (\text{t.g. kiểm tra } j \leq n-i)\right] = \sum_{j=1}^{n-i}\left[t_3^j + O(1)\right]$$

t_4 là $O(1)$ $(v := a_j,\ a_j := a_{j+1},\ a_{j+1} := v)$

$t_3^j = t_4 + (\text{thời gian kiểm tra } a_j > a_{j+1})$

$\boxed{T(n) \text{ là } O(n^2)}$

(xấu nhất)

$\boxed{T(n) \text{ là } O(n^2)}$

(tốt nhất)

Độ phức tạp tính toán

Sắp xếp chèn

Thuật toán 5: Sắp xếp chèn (Insertion Sort)

Input: $a_1, a_2, \ldots, a_n$: dãy số thực ($n \geq 2$)
Output: Dãy đã cho được sắp xếp theo thứ tự tăng dần

for $i = 2$ **to** n **do** — t_1
    $m := a_i$ — t_2^i
    $j := i - 1$ — t_3^i
    while $j \geq 1$ và $m < a_j$ **do** — t_4^i
        $a_{j+1} := a_j$ — $t_6^{i,j}$
        $j := j - 1$
    $a_{j+1} := m$ — t_5^i

$$T(n) = t_1 = \sum_{i=2}^{n} \left[(t_2^i + t_3^i + t_4^i + t_5^i) + (\text{t.g. kiểm tra } i \leq n) \right]$$

$t_2^i, t_3^i, t_5^i, t_6^{i,j}$ là $O(1)$

$$t_4^i = \sum_{1 \leq j \leq i-1 \text{ và } m < a_j} \left[t_6^{i,j} + (\text{t.g. kiểm tra } j \geq 1 \text{ và } m < a_j) \right]$$

$\boxed{T(n) \text{ là } O(n^2)}$ (xấu nhất)

$\boxed{T(n) \text{ là } O(n)}$ (tốt nhất)

Thuật toán tham lam
Giới thiệu

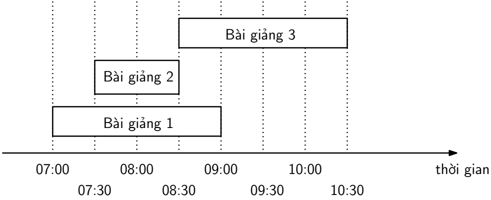
- *Các bài toán tối ưu (optimization problems)* yêu cầu cực đại hóa hoặc cực tiểu hóa một số tham số xét trên tập tất cả các đầu vào có thể
  - Tìm đường đi giữa hai thành phố với khoảng cách *nhỏ nhất*
  - Tìm cách mã hóa các thông điệp sử dụng số lượng bit *nhỏ nhất* có thể
- Một *thuật toán tham lam (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.

Thuật toán tham lam
Lập lịch

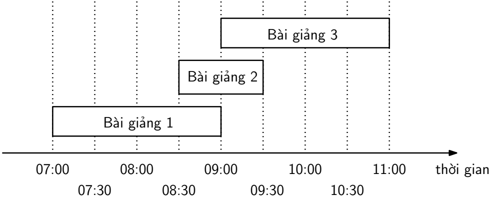
- **Bài toán:**
  - **Input:**
    - Một nhóm các bài giảng với thời gian bắt đầu và kết thúc
    - Chỉ có một giảng đường duy nhất
    - Khi một bài giảng bắt đầu, nó tiếp diễn cho đến khi kết thúc
    - Không có hai bài giảng nào được tiến hành ở cùng thời điểm.
    - Ngay sau khi một bài giảng kết 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?
  - Chọn bài giảng bắt đầu sớm nhất trong số các bài giảng bắt đầu sau các bài giảng ta vừa chọn trước đó?
  - Chọn bài giảng ngắn nhất trong số các bài giảng bắt đầu sau các bài giảng ta vừa chọn trước đó?
  - Chọn bài giảng kết thúc sớm nhất trong số các bài giảng bắt đầu sau các bài giảng ta vừa chọn trước đó?

Thuật toán tham lam
Lập lịch

- Chọn bài giảng bắt đầu sớm nhất? *Không xuất ra lời giải tối ưu trong mọi trường hợp*



- Chọn bài giảng ngắn nhất? *Không xuất ra lời giải tối ưu trong mọi trường hợp*



Thuật toán 6: 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

Sắp xếp các bài giảng theo thứ tự tăng dần theo thời gian kết thúc và gán lại nhãn bài giảng sao cho $e_1 \leq e_2 \leq \cdots \leq e_n$

$S := \emptyset$

for $j := 1$ **to** n **do**

  if *Bài giảng j không xung đột với các phần tử của S* **then**

    $S := S \cup \{j\}$

return S

Thuật toán II
Hoàng Anh Đức

Định lý 1

Thuật toán 6 xuất ra một danh sách các bài giảng tối ưu

Chứng minh.

- Giả sử $S^\star$ là một danh sách bài giảng tối ưu trong đó các bài giảng $x_1, x_2, \dots, x_k$ được sắp xếp theo thứ tự tăng dần theo thời gian kết thúc
- Giả sử S là một danh sách bài giảng xuất ra từ Thuật toán 6 trong đó các bài giảng $y_1, y_2, \dots, y_{k'}$ được sắp xếp theo thứ tự tăng dần theo thời gian kết thúc
- Do $S^\star$ là tối ưu, $k \geq k'$
- Nếu $S = S^\star$ 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

$$S^\star = \langle x_1 = y_1, x_2 = y_2, \dots, x_{i-1} = y_{i-1}, x_i, \dots, x_{k'}, \dots, x_k \rangle$$
$$S = \langle x_1 = y_1, x_2 = y_2, \dots, x_{i-1} = y_{i-1}, y_i, \dots, y_{k'} \rangle$$

Chứng minh (tiếp).

- Nếu i không tồn tại thì $S^\star = \langle y_1, y_2, \dots, y_{k'}, x_{k'+1}, \dots, x_k \rangle$
- Ngược lại, do y_i được chọn bởi Thuật toán 6, y_i kết thúc trước khi x_i kết thúc, nghĩa là nó không xung đột với bất kỳ bài giảng nào sau x_i trong $S^\star$.
- Do đó, dãy $S^\star \setminus \{x_i\} \cup \{y_i\}$ cũng là một dãy tối ưu. Ta gán $S^\star := S^\star \setminus \{x_i\} \cup \{y_i\}$ và lặp lại lý luận trên cho $S^\star$ và S.
- Bằng cách liên tục sử dụng lý luận trên, ta thu được dãy $S^\star$ tối ưu có dạng

$$S^\star = \langle y_1, y_2, \dots, y_{k'}, x_{k'+1}, \dots, x_k \rangle$$

- Do $y_{k'}$ là bài giảng cuối cùng được chọn bởi Thuật toán 6, các bài giảng còn lại đều xung đột với $y_{k'}$ và có thời gian kết thúc sau khi $y_{k'}$ kết thúc. Nói cách khác, các phần tử $x_{k'+1}, \dots, x_k$ trong $S^\star$ đều phải bằng $y_{k'}$, nghĩa là $S = \langle y_1, y_2, \dots, y_{k'} \rangle$ là một dãy tối ưu

□

- Một ứng dụng của việc Giải hệ thức truy hồi mà chúng ta sẽ đề cập ở phần sau là trong việc phân tích độ phức tạp của các thuật toán đệ quy
- 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
- Một số phương pháp ước lượng hệ thức truy hồi
  (1) Sử dụng cây đệ quy
  (2) Sử dụng định lý thợ (Master Theorem)

Giải hệ thức truy hồi

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 $g_n = ng_{n-1}$, $g_n = g_{n-1} + g_{n-2}^2$ không là hệ thức truy hồi tuyến tính thuần nhất với hệ số hằng

Giải hệ thức truy hồi
Đa thức đặc trưng

- Hệ thức $a_n = c_1 a_{n-1} + \cdots + c_k a_{n-k}$ $(c_k \neq 0)$ 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, $r^n = c_1 r^{n-1} + \cdots + c_k a r^{n-k}$. Chia hai vế cho r^{n-k} và chuyển vế, ta có

    $$r^k - c_1 r^{k-1} - c_2 r^{k-2} - \cdots - c_{k-1} r - c_k = 0.$$

    Đa thức cuối cùng gọi là *đa thức đặc trưng (characteristic equation)* 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

Định lý 2

Cho các số thực c_1, c_2. Giả sử $r^2 - c_1 r - c_2$ có hai nghiệm phân biệt r_1 và r_2. Dãy $\{a_n\}$ là nghiệm của hệ thức truy hồi $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ $(n \geq 2)$ với điều kiện ban đầu $a_0 = C_0$ và $a_1 = C_1$ khi và chỉ khi $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ với mọi $n \geq 0$, trong đó α_1, α_2 là các hằng số nào đó

Chứng minh sơ lược.

Ta cần chứng minh hai điều

(1) Nếu r_1, r_2 là các nghiệm của đa thức đặc trưng, thì $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ với mọi $n \geq 0$, trong đó α_1, α_2 là các hằng số nào đó, thỏa mãn hệ thức truy hồi

(2) Nếu $\{a_n\}$ là nghiệm của hệ thức truy hồi với các điều kiện ban đầu $a_0 = C_0$ và $a_1 = C_1$, thì tồn tại các hằng số α_1, α_2 sao cho $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$

Tương tự, ta có

Định lý 3

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 k nghiệm phân biệt $r_1, r_2, \dots, r_k$. 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 $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \dots + \alpha_k r_k^n$ với mọi $n \geq 0$, trong đó $\alpha_1, \alpha_2, \dots, \alpha_k$ là các hằng số nào đó

Tổng quát cho trường hợp đa thức đặc trưng có nghiệm bội

Định lý 4

Cho các số thực $c_1, c_2, \ldots, c_k$. *Giả sử* $r^k - c_1 r^{k-1} - \cdots - c_{k-1} r - c_k$ *có* t *nghiệm phân biệt* $r_1, r_2, \ldots, r_t$ *với các bội tương ứng* $m_1, m_2, \ldots, m_t$ *thỏa mãn* $m_i \geq 1$ *với* $1 \leq i \leq t$ *và* $m_1 + \cdots + m_t = k$ *(nghĩa là, có* m_i *nghiệm có giá trị* r_i*). Dãy* $\{a_n\}$ *là nghiệm của hệ thức truy hồi* $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$ $(n \geq k)$ *với điều kiện ban đầu* $a_0 = C_0, a_1 = C_1, \ldots, a_{k-1} = C_{k-1}$ *khi và chỉ khi*

$$\begin{aligned} a_n &= (\alpha_{1,0} + \alpha_{1,1} n + \cdots + \alpha_{1,m_1-1} n^{m_1-1}) r_1^n \\ &\quad + (\alpha_{2,0} + \alpha_{2,1} n + \cdots + \alpha_{2,m_2-1} n^{m_2-1}) r_2^n \\ &\quad + \cdots + (\alpha_{t,0} + \alpha_{t,1} n + \cdots + \alpha_{t,m_t-1} n^{m_t-1}) r_t^n \end{aligned}$$

với $n \geq 0$, *trong đó* $\alpha_{i,j}$ *là các hằng số với* $1 \leq i \leq t$ *và* $0 \leq j \leq m_i - 1$

Ví dụ 1

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 $r^2 - 2r - 1$
- 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 $\alpha_2 = -1/\sqrt{5}$

Ví dụ 2

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 $a_n = \alpha_{1,0} r_1^n + \alpha_{1,1} n r_1^n = \alpha_{1,0} 3^n + \alpha_{1,1} n 3^n$ 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 $\alpha_{1,1} = 1$

Ví dụ 3

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 $a_n = \alpha_{1,0}r_1^n + \alpha_{1,1}nr_1^n + \alpha_{1,2}n^2r^n$
- 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

$$
\begin{aligned}
a_0 &= \alpha_{1,0} = 1 \\
a_1 &= -\alpha_{1,0} - \alpha_{1,1} - \alpha_{1,2} = -2 \\
a_2 &= \alpha_{1,0} + 2\alpha_{1,1} + 4\alpha_{1,2} = -1
\end{aligned}
$$

Do đó ta có $\alpha_{1,0} = 1$, $\alpha_{1,1} = 3$, và $\alpha_{1,2} = -2$

Giải hệ thức truy hồi

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 $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$

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 $n^m (p_t n^t + p_{t-1} n^{t-1} + \cdots + p_1 n + p_0) s^n$.

Ví dụ 4

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 $c = 0$

Bài tập 1

Giải các hệ thức truy hồi với điều kiện ban đầu sau

(1) $a_n = 2a_{n-1}$ *với* $n \geq 1$, $a_0 = 3$

(2) $a_n = 5a_{n-1} - 6a_{n-2}$ *với* $n \geq 2$, $a_0 = 1$, $a_1 = 0$

(3) $a_n = 4a_{n-1} - 4a_{n-2}$ *với* $n \geq 2$, $a_0 = 6$, $a_1 = 8$

(4) $a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$ *với* $n \geq 3$, $a_0 = 5$, $a_1 = -9$, $a_2 = 15$

(5) $a_n = 3a_{n-1} + 2^n$ *với* $n \geq 1$, $a_0 = 1$

(6) $a_n = 2a_{n-1} + 2n^2$ *với* $n \geq 2$, $a_1 = 5$

(7) $a_n = 5a_{n-1} - 6a_{n-2} + 2^n + 3n$ *(**Gợi ý:** Tìm nghiệm riêng có dạng* $qn2^n + p_1 n + p_2$*, trong đó* q, p_1, p_2 *là các hằng số)*

Hàm sinh

Hàm sinh (generating function) $G_a(x)$ của một dãy vô hạn $\{a_n\}$ $(n \geq 0)$ được định nghĩa như sau

$$G_a(x) := \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k + \ldots$$

Nói cách khác, a_n là hệ số của x^n trong $G_a(x)$

$G_a(x) = \sum_{n=0}^{\infty} a_n x^n$

$a_n = f(a_{n-1}, a_{n-2}, \ldots)$

$a_0 = C_0, a_1 = C_1, \ldots.$

Công thức tường minh

$G_a(x) = h(x)$

Khai triển $h(x)$

$h(x) = \sum_{n=0}^{\infty} \boxed{}\, x^n$

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

Ví dụ 5

Giải hệ thức truy hồi $a_n = 3a_{n-1}$ $(n \geq 1)$ với điều kiện ban đầu $a_0 = 2$

$$\begin{aligned} G_a(x) &= \sum_{n=0}^{\infty} a_n x^n = a_0 + \sum_{n=1}^{\infty} (3a_{n-1})x^n \\ &= 2 + 3x \sum_{n=1}^{\infty} a_{n-1}x^{n-1} = 2 + 3x \sum_{m=0}^{\infty} a_m x^m \\ &= 2 + 3xG_a(x) \\ G_a(x) &= \frac{2}{1-3x} \end{aligned}$$

Nhắc lại rằng $\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$ với $-1 < x < 1$. Suy ra $\sum_{n=0}^{\infty} c^n x^n = \frac{1}{1-cx}$ với $-1 < cx < 1$ trong đó $c \neq 0$ 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 $a_n = 2 \cdot 3^n$

Giải hệ thức truy hồi
Hàm sinh

Một phương pháp khác để tìm công thức tường minh cho $G_a(x)$ trong Ví dụ 5. Chú ý rằng $a_n = 3a_{n-1}$ $(n \geq 1)$ và $a_0 = 2$

$$G_a(x) = \sum_{n=0}^{\infty} a_n x^n \qquad \text{Định nghĩa hàm sinh}$$

$$-3xG_a(x) = -\sum_{n=0}^{\infty} [3a_n] x^{n+1} \qquad \text{Nhân hai vế với } -3x$$

$$\begin{aligned}
G_a(x) - 3xG_a(x) &= \sum_{n=0}^{\infty} a_n x^n - \sum_{n=0}^{\infty} [3a_n] x^{n+1} && \text{Cộng hai đẳng thức} \\
&= a_0 x^0 + \sum_{n=1}^{\infty} a_n x^n - \sum_{n=0}^{\infty} [3a_n] x^{n+1} \\
&= 2 + \sum_{n=1}^{\infty} a_n x^n - \sum_{n=1}^{\infty} [3a_{n-1}] x^n && \text{Đổi chỉ số} \\
&= 2 + \sum_{n=1}^{\infty} \big[a_n - 3a_{n-1}\big] x^n && a_n = 3a_{n-1} \\
&= 2
\end{aligned}$$

Một số đẳng thức hữu ích

$$\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n$$

$$\frac{1}{1-ax} = \sum_{n=0}^{\infty} a^n x^n$$

$$\frac{1}{1-x^k} = \sum_{n=0}^{\infty} x^{kn}$$

$$\frac{1}{(1-x)^2} = \sum_{n=0}^{\infty} (n+1) x^n$$

Ví dụ 6

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 $f_1 = 1$

$$\begin{aligned}
G_f(x) &= \sum_{n=0}^{\infty} f_n x^n = f_0 + f_1 x + \sum_{n=2}^{\infty} (f_{n-1} + f_{n-2}) x^n \\
&= x + x \sum_{n=2}^{\infty} f_{n-1} x^{n-1} + x^2 \sum_{n=2}^{\infty} x^{n-2} \\
&= x + x \sum_{m=1}^{\infty} f_m x^m + x^2 \sum_{m=0}^{\infty} f_m x^m && \text{đổi biến} \\
&= x + x[\sum_{m=0}^{\infty} f_m x^m - f_0 x^0] + x^2 \sum_{m=0}^{\infty} f_m x^m && f_0 x^0 = 0 \\
&= x + x G_f(x) + x^2 G_f(x)
\end{aligned}$$

$$G_f(x) = \frac{x}{1 - x - x^2}$$

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 2

- *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* $\{f_n\}$

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 3

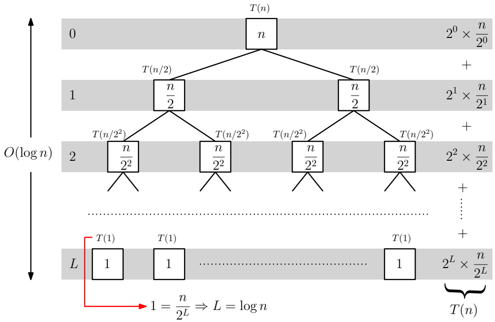
Giải hệ thức truy hồi $a_n = 2a_{n-1} - a_{n-2}$ với $n \geq 2$ và các điều kiện ban đầu $a_0 = 1$, $a_1 = 1$ bằng cách sử dụng hàm sinh

Bài tập 4 ($\star$)

Giải hệ thức truy hồi $a_n = 3a_{n-1} + n$ với $n \geq 1$ và điều kiện ban đầu $a_0 = 1$ bằng cách sử dụng hàm sinh

Ví dụ 7

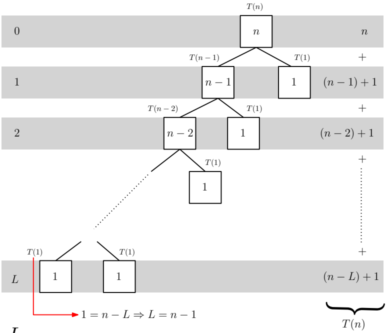
Xét hệ thức truy hồi $T(n) = 2T(n/2) + n$ với điều kiện ban đầu $T(1) = 1$ và $n = 2^k$ với số nguyên $k \geq 1$ nào đó. Ta vẽ cây đệ quy cho hệ thức này như sau



Ta có $T(n) = \sum_{i=0}^{L} 2^i \times \frac{n}{2^i} = n(\log n + 1) = O(n \log n)$

Ví dụ 8

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



Ta có $T(n) \leq \sum_{i=0}^{L} n = O(n^2)$

Định lý 8: 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 9

- 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 $T(n) = O(n^{\log 3})$

- 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

Một thuật toán đệ quy thường có dạng như sau

Trường hợp cơ sở Với một số đầu vào kích thước nhỏ hoặc một số trường hợp đặc biệt, thuật toán sẽ cho ra kết quả một cách trực tiếp

Định nghĩa bài toán con Trong trường hợp đầu vào khác với những đầu vào định nghĩa trong trường hợp cơ sở, thuật toán định nghĩa một hoặc nhiều "bài toán con" với các đầu vào nhỏ hơn được tính từ đầu vào ban đầu

Giải bài toán con Thuật toán gọi chính nó để giải các bài toán con và lưu trữ các kết quả tính toán

Xuất ra lời giải Sau khi giải quyết toàn bộ các bài toán con, thuật toán xuất ra lời giải dựa trên đầu vào ban đầu và các lời giải của các bài toán con

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

Với mọi số nguyên không âm n

$$0! = 1$$
$$n! = n \times (n-1)! \quad \forall n \geq 1$$

Thuật toán 7: Tính $n!$

Input: n: số nguyên không âm
Output: $n!$

```
procedure factorial(n):
    if n = 0 then
        return 1
    else
        return n × factorial(n − 1)
```

Ví dụ 10 (Thuật toán đệ quy tính giai thừa là đúng)

Ta chứng minh tính đúng đắn của Thuật toán 7 bằng quy nạp mạnh. Gọi $\texttt{factorial}(n)$ là giá trị trả lại bởi Thuật toán 7

- Ta chứng minh $\texttt{factorial}(n) = n!$ với mọi $n \geq 0$
- **Bước cơ sở:** Khi $n = 0$, $\texttt{factorial}(n) = 1 = n!$
- **Bước quy nạp:** Giả sử $\texttt{factorial}(j) = j!$ với mọi j thỏa mãn $0 \leq j \leq k$ với số nguyên $k \geq 0$ nào đó. Ta chứng minh $\texttt{factorial}(k+1) = (k+1)!$. Thật vậy, Thuật toán 7 trả lại $\texttt{factorial}(k+1) = (k+1) \times \texttt{factorial}(k)$. Theo giả thiết quy nạp, $\texttt{factorial}(k) = k!$. Do đó, $\texttt{factorial}(k+1) = (k+1) \times k! = (k+1)!$

Ví dụ 11 (Thời gian chạy của thuật toán đệ quy tính giai thừa)

$$T(n) = \max\{O(1), T(n-1) + O(1)\} + O(1) = T(n-1) + O(1)$$

Thuật toán đệ quy

Tính lũy thừa

Với mọi số thực $a \neq 0$ và số nguyên không âm n,

$$a^0 = 1$$
$$a^n = a \times a^{n-1} \quad \forall n \geq 1$$

Thuật toán 8: 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 × power(a, n − 1)
```

Thuật toán 9: Tìm kiếm tuyến tính (Linear Search)

Input: $a_1, a_2, \dots, a_n$: dãy số nguyên, i, j, x: số nguyên, $1 \le i \le j \le n$

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

```
procedure LinearSearch(i, j, x):
    if a_i = x then   // Ở đúng vị trí? Trả lại kết quả
        return i
    else
        if i = j then                    // Không tìm thấy
            return 0
        else                   // Tìm trong phần còn lại
            return LinearSearch(i + 1, j, x)
```

Thuật toán đệ quy

Tìm kiếm nhị phân

Thuật toán 10: Tìm kiếm nhị phân (Binary Search)

Input: $a_1, a_2, \ldots, a_n$: dãy số nguyên thực sự tăng, i, j, x: số nguyên, $1 \leq i \leq j \leq n$

Output: Nếu $x \in \{a_i, a_{i+1}, \ldots, a_j\}$ thì trả lại $k \in \{i, i+1, \ldots, j\}$ sao cho $x = a_k$. Ngược lại thì trả lại 0

procedure `BinarySearch`(i, j, x):

- $m := \lfloor (i+j)/2 \rfloor$ — `// Đi đến giữa dãy`
- **if** $x = a_m$ **then** — `// Đúng vị trí?`
  - **return** m
- **else**
  - **if** $x < a_m$ *và* $i < m$ **then** — `// x ở nửa bên trái?`
    - **return** `BinarySearch`$(i, m-1, x)$
  - **else**
    - **if** $x > a_m$ *và* $j > m$ **then** — `// x ở nửa bên phải?`
      - **return** `BinarySearch`$(m+1, j, x)$
    - **else** — `// Không tìm thấy`
      - **return** 0

Thuật toán 11: 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

procedure `MergeSort`(L):
  if $n > 1$ **then**
    $m := \lfloor n/2 \rfloor$
    $L_1 := a_1, \dots, a_m$
    $L_2 := a_{m+1}, \dots, a_n$
    $L :=$ `Merge`(`MergeSort`(L_1), `MergeSort`(L_2))

Thuật toán đệ quy
Sắp xếp trộn

Thuật toán 12: Trộn hai dãy sắp thứ tự (Merge)

Input: $A = (a_1, \ldots, a_{|A|}), B = (b_1, \ldots, b_{|B|})$: dãy số nguyên sắp thứ tự

Output: Dãy các số nguyên trong cả A và B sắp thứ tự tăng dần

procedure `Merge`(A, B):
    if $A = \emptyset$ **then**
        return B
    if $B = \emptyset$ **then**
        return A
    if $a_1 < b_1$ **then**
        return $(a_1, \texttt{Merge}(a_2, \ldots, a_{|A|}, B))$
    else
        return $(b_1, \texttt{Merge}(A, b_2, \ldots, b_{|B|}))$

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

- *Lý thuyết số (number theory)* nghiên cứu các tính chất và mối liên hệ giữa các loại số
  - quan trọng nhất là *các số nguyên dương (positive integers)*
  - đặc biệt là *các số nguyên tố (prime numbers)*

- Cho các số nguyên a và b với $a \neq 0$. Ta nói b *chia hết cho* a, ký hiệu $b \vdots a$, 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 $b \not\vdots\ a$ 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

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?)

- 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 $q = a \text{ div } d$ và $r = a \bmod d$. Chú ý rằng với d cố định, $a \text{ div } d$ 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

- $101 \text{ div } 11 = 9$ và $101 \bmod 11 = 2$
- $-11 \text{ div } 3 = \ -4$ và $-11 \bmod 3 = 1$
  (Chú ý rằng mặc dù $-11 = 3(-3) - 2$ nhưng *số dư của phép chia $a = -11$ cho $d = 3$ không bằng -2* do $r = -2$ không thỏa mãn $0 \leq r < d$)

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

procedure `div-mod`(a, d):

- $q := 0$
- $r := |a|$
- **while** $r \geq d$ **do** // Tiếp tục trừ d từ r và tăng q cho đến khi $r < d$
  - $r := r - d$
  - $q := q + 1$
- **if** $a < 0$ *và* $r > 0$ **then** // Trường hợp a âm
  - $r := d - r$
  - $q := -(q + 1)$
- **return** (q, r) // $q = a$ div d là thương, $r = a \bmod d$ là số dư

- 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 $a \equiv b \pmod{m}$. Theo định nghĩa, $m \mid (a-b)$. Nếu $a = q_1 m + r_1$ và $b = q_2 m + r_2$ với $0 \leq r_1 < m$ và $0 \leq r_2 < m$ thì $a - b = (q_1 - q_2)m + (r_1 - r_2)$. Do $0 \leq r_1, r_2 < m$ nên $-m < r_1 - r_2 < m$. Do $m \mid (a-b)$ nên $r_1 - r_2 = mp$ với $p \in \mathbb{Z}$. Suy ra $-m < mp < m$ và do đó $p = 0$, nghĩa là $r_1 = r_2$, hay nói cách khác $a \bmod m = b \bmod m$

($\Leftarrow$) Giả sử $a \bmod m = b \bmod m = r$. Suy ra $a = q_1 m + r$ và $b = q_2 m + r$. Do đó, $a - b = (q_1 - q_2)m$, nghĩa là $m \mid (a-b)$ $\square$

Bài tập 2

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, $a \equiv b \pmod m$ $\square$

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

- $(a + b) \bmod m = ((a \bmod m) + (b \bmod m)) \bmod m$
- $ab \bmod m = ((a \bmod m)(b \bmod m)) \bmod m$

- Ta có thể định nghĩa các toán tử số học trên tập $\mathbb{Z}_m = \{0, 1, \dots, m-1\}$: Với $a, b \in \mathbb{Z}_m$
  - $a +_m b = (a + b) \bmod m$; và
  - $a \cdot_m b = (a \cdot b) \bmod m$,

  trong đó các phép toán $+$ và $\cdot$ ở vế phải là các phép toán trên $\mathbb{Z}$. Các phép toán $+_m$ và $\cdot_m$ được gọi là các phép cộng và nhân theo môđun m

Biểu diễn số nguyên

Biểu diễn theo hệ b-phân

- Thông thường, chúng ta biểu diễn các số theo *hệ cơ số (base)* 10, sử dụng các *chữ số (digit)* từ 0 đến 9
- Trên thực tế, ta có thể biểu diễn các số theo hệ cơ số $b > 1$ bất kỳ
- Với mọi $n, b \in \mathbb{Z}^+$, tồn tại duy nhất một dãy $a_k a_{k-1} \dots a_1 a_0$ gồm các *chữ số* $a_i < b$ $(1 \leq i \leq k)$ thỏa mãn

  $$n = a_k b^k + a_{k-1} b^{k-1} + a_{k-2} b^{k-2} + \cdots + a_1 b^1 + a_0 = \sum_{i=0}^{k} a_i b^i$$

  Ta cũng ký hiệu $n = (a_k a_{k-1} \dots a_2 a_1)_b$
- 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)

Ví dụ 2

$$(101011111)_2 = (?)_{10} 1\cdot 2^8 + 0\cdot 2^7 + 1\cdot 2^6 + 0\cdot 2^5 + 1\cdot 2^4 + 1\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = (351)_{10}$$

$$(2AE0B)_{16} = (?)_{10} 2\cdot 16^4 + 10\cdot 16^3 + 14\cdot 16^2 + 0\cdot 16^1 + 11\cdot 16^0 = (175627)_{10}$$

Để chuyển một số nguyên n sang hệ b phân với $b > 1$:

(1) Để tìm giá trị của chữ số ngoài cùng bên phải, tính $n \bmod b$

(2) Thay n bởi $n \operatorname{div} b$

(3) Lặp lại các bước (1) và (2) cho đến khi $n = 0$

Bài tập 3

Mô tả thuật toán trên bằng mã giả

Chuyển một số nguyên n sang hệ b phân với $b > 1$:

$$
\begin{aligned}
n &= bq_0 + a_0 && n := q_0 \\
&= b(bq_1 + a_1) + a_0 \\
&= b^2 q_1 + ba_1 + a_0 && n := q_1 \\
&= b^2(bq_2 + a_2) + ba_1 + a_0 \\
&= b^3 q_2 + b^2 a_2 + ba_1 + a_0 && n := q_2 \\
&= b^3(bq_3 + a_3) + b^2 a_2 + ba_1 + a_0 \\
&= b^4 q_3 + b^3 a_3 + b^2 a_2 + ba_1 + a_0 && n := q_3 \\
&\vdots && \vdots \\
&= b^k(0 + a_k) + b^{k-1} a_{k-1} + \ldots b^3 a_3 + b^2 a_2 + ba_1 + a_0 && n := 0 \\
&= b^k a_k + b^{k-1} a_{k-1} + \ldots b^3 a_3 + b^2 a_2 + ba_1 + a_0
\end{aligned}
$$

Ví dụ 3

$$(12345)_{10} = (?)_8$$

$$12345 = 8 \cdot 1543 + 1$$
$$1543 = 8 \cdot 192 + 7$$
$$192 = 8 \cdot 24 + 0$$
$$24 = 8 \cdot 3 + 0$$
$$3 = 8 \cdot 0 + 3$$

Do đó, $(12345)_{10} = (30071)_8$

Biểu diễn số nguyên

Chuyển đổi giữa các hệ nhị phân, bát phân, và thập lục phân

Chuyển đổi giữa hệ nhị phân và hệ bát phân (hoặc hệ thập lục phân) rất dễ thực hiện

- 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

$$(\underline{11}\ \underline{1110}\ \underline{1011}\ \underline{1111})_2 = (3EBF)_{16}$$

$$(\underline{A}\underline{3}\underline{D})_{16} = (1010\ 0011\ 1101)_2$$

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 $a + b = (s_ns_{n-1}\dots s_1s_0)_2$

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 $s = a + b$

procedure `add`(a, b):

- $c := 0$
- **for** $j := 0$ **to** $n - 1$ **do**
  - $d := \lfloor (a_j + b_j + c)/2 \rfloor$
  - $s_j = a_j + b_j + c - 2d$
  - $c := d$
- $s_n := c$
- **return** $(s_0, s_1, \dots, s_n)$

Ví dụ 4

Cộng hai số $a = (10111)_2$ và $b = (11100)_2$

$$\begin{array}{r} 10111 \\ +\;11100 \\ \hline 1 \end{array} \quad \text{nhớ } 0 \qquad \begin{array}{r} 10111 \\ +\;11100 \\ \hline 11 \end{array} \quad \text{nhớ } 0 \qquad \begin{array}{r} 10111 \\ +\;11100 \\ \hline 011 \end{array} \quad \text{nhớ } 1$$

$$\begin{array}{r} 10111 \\ +\;11100 \\ \hline 0011 \end{array} \quad \text{nhớ } 1 \qquad \begin{array}{r} 10111 \\ +\;11100 \\ \hline 110011 \end{array} \quad \text{nhớ } 1$$

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

$$\begin{aligned} ab &= a(b_0 2^0 + b_1 2^1 + \cdots + b_{n-1}2^{n-1}) \\ &= a(b_0 2^0) + a(b_1 2^1) + \cdots + a(b_{n-1}2^{n-1}) \end{aligned}$$

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 $j \in \{0, \dots, n-1\}$

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 $p = ab$

procedure `multiply`(a, b):
    for $j := 0$ **to** $n-1$ **do**
        if $b_j = 1$ **then**
            $c_j := a$ sau khi di chuyển j đơn vị sang trái
        else
            $c_j := 0$
        `//` $c_0, \dots, c_{n-1}$ `là các tích thành phần`
        $p := 0$
        for $j := 0$ **to** $n-1$ **do**
            $p :=$ `add`(p, c_j)
    return p

Ví dụ 5

Nhân hai số $a = (110)_2$ và $b = (101)_2$

$$\begin{array}{r} 110 \\ \times\ 101 \\ \hline 110 \end{array} \qquad \begin{array}{r} 110 \\ \times\ 101 \\ \hline 110 \\ 0000 \end{array} \qquad \begin{array}{r} 110 \\ \times\ 101 \\ \hline 110 \\ +\ 0000 \\ 11000 \\ \hline 11110 \end{array}$$

- Trong hệ nhị phân, các số âm có thể được biểu diễn thông qua *ký hiệu phần bù hai (two's complement notation)*
- Trong trường hợp này, một chuỗi nhị phân n bit có thể biểu diễn bất kỳ số nguyên i nào thỏa mãn $-2^{n-1} \leq i < 2^{n-1}$
- 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 $a = (a_{n-1} \ldots a_0)_2$ thì $-a = (\overline{a_{n-1} \ldots a_0})_2 + 1$, trong đó $\overline{a_{n-1} \ldots a_0}$ là phần bù của $a_{n-1} \ldots a_0$ thu được thông qua tính toán bằng toán tử lôgic $\overline{}$ (phủ định) theo từng bit

Ví dụ 6 (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

Tính chia hết và phép toán môđun

Tính lũy thừa môđun

- 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*

- Chú ý rằng

$$b^n = b^{\overbrace{a_{k-1}2^{k-1}+a_{k-2}2^{k-2}+\cdots+a_1 2^1+a_0 2^0}^{\text{Biểu diễn nhị phân của } n}}$$
$$= (b^{2^{k-1}})^{a_{k-1}} \times (b^{2^{k-2}})^{a_{k-2}} \times \cdots \times (b^{2^1})^{a_1} \times (b^{2^0})^{a_0}$$

- 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 $\bmod m$ của kết quả để tiếp tục thực hiện tính toán*

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: $b^n \bmod m$

$x := 1$ // để lưu trữ kết quả

$b2i := b \bmod m$ // b^{2^i}, đầu tiên $i = 0$

for $i := 0$ **to** $k - 1$ **do** // xét tất cả k bit của n

  if $a_i = 1$ **then**

    $x := (x \cdot b2i) \bmod m$

  $b2i := (b2i \cdot b2i) \bmod m$ // $b^{2^{i+1}} = (b^{2^i}) \cdot (b^{2^i})$

return x

- 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: $2, 3, 5, 11, \ldots$
- Các số nguyên lớn hơn 1 và không phải là số nguyên tố được gọi là các *hợp số (composite number)*

Bài tập 4

*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 5

Sử dụng quy nạp, hãy chứng minh phát biểu tổng quát: nếu p là một số nguyên tố và $p \mid a_1 a_2 \ldots a_n$, trong đó $a_i \in \mathbb{Z}$ với $1 \leq i \leq n$, thì $p \mid a_j$ với j nào đó ($1 \leq j \leq n$)

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 $n = p_1p_2 \dots p_s$ và $n = q_1q_2 \dots q_t$, 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 $p_1 \leq p_2 \leq \cdots \leq p_s$ và $q_1 \leq q_2 \leq \cdots \leq q_t$. Sử dụng Bài tập 5 để chỉ ra mâu thuẫn

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 $\sqrt{n}$

Số nguyên tố và Ước chung lớn nhất
Số nguyên tố

- 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)*
  Thử mọi số nguyên i thỏa mãn $2 \leq i \leq \sqrt{n}$ và kiểm tra xem n có chia hết cho i không
  (1) Viết các số $2, \ldots, n$ 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)

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

Số nguyên tố và Ước chung lớn nhất

Ước chung lớn nhất

- 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

$$a = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n} \qquad b = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n}$$

trong đó các số mũ là các số nguyên không âm (có thể bằng 0), thì

$$\gcd(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \dots p_n^{\min(a_n, b_n)}$$

Số nguyên tố và Ước chung lớn nhất

Bội chung nhỏ nhất và liên hệ với Ước chung lớn nhất

- Cho $a, b \in \mathbb{Z}^+$. *Bội chung nhỏ nhất (least common multiple)* của a và b, ký hiệu $\operatorname{lcm}(a, b)$, 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

$$a = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n} \qquad b = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n}$$

trong đó các số mũ là các số nguyên không âm (có thể bằng 0), thì

$$\operatorname{lcm}(a, b) = p_1^{\max(a_1, b_1)} p_2^{\max(a_2, b_2)} \dots p_n^{\max(a_n, b_n)}$$

Định lý 10

Với $a, b \in \mathbb{Z}^+$, $ab = \gcd(a, b) \cdot \operatorname{lcm}(a, b)$

Bài tập 7

Chứng minh Định lý 10

Bổ đề 11

Cho $a = bq + r$ *với* a, b, q, r *là các số nguyên. Ta có* $\gcd(a, b) = \gcd(b, r)$. *Nói cách khác* $\gcd(a, b) = \gcd(b, (a \bmod b))$

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 $\gcd(a, b) = \gcd(b, r)$

□

Ví dụ 7

Tìm $\gcd(372, 164)$

- $\gcd(372, 164) = \gcd(164, 372 \bmod 164)$
  - $372 \bmod 164 = 372 - 164\lfloor 372/164 \rfloor = 372 - 164 \cdot 2 = 44$
- $\gcd(164, 44) = \gcd(44, 164 \bmod 44)$
  - $164 \bmod 44 = 164 - 44\lfloor 164/44 \rfloor = 164 - 44 \cdot 3 = 32$
- $\gcd(44, 32) = \gcd(32, 44 \bmod 32)$
  - $44 \bmod 32 = 44 - 32\lfloor 44/32 \rfloor = 44 - 32 \cdot 1 = 12$
- $\gcd(32, 12) = \gcd(12, 32 \bmod 12)$
  - $32 \bmod 12 = 32 - 12\lfloor 32/12 \rfloor = 32 - 12 \cdot 2 = 8$
- $\gcd(12, 8) = \gcd(8, 12 \bmod 8)$
  - $12 \bmod 8 = 12 - 8\lfloor 12/8 \rfloor = 12 - 8 \cdot 1 = 4$
- $\gcd(8, 4) = \gcd(4, 8 \bmod 4)$
  - $8 \bmod 4 = 8 - 4\lfloor 8/4 \rfloor = 0$
- $\gcd(4, 0) = 4$

Thuật toán 5: Thuật toán Euclid

Input: a, b: các số nguyên dương
Output: $\gcd(a, b)$

```
x := a
y := b
while y ≠ 0 do
    r := x mod y
    x := y
    y := r
return x                                  // x = gcd(a, b)
```

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)

Ví dụ 8

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

$$\begin{aligned} 18 &= 54 - 1 \cdot 36 \\ &= 54 - 1 \cdot (198 - 3 \cdot 54) \\ &= 4 \cdot 54 - 1 \cdot 198 \\ &= 4 \cdot (252 - 1 \cdot 198) - 1 \cdot 198 \\ &= 4 \cdot 252 - 5 \cdot 198 \end{aligned}$$

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)
    (d1, s1, t1) := ExtEuclid(b, a mod b)
    d := d1
    s := t1
    t := s1 − (a div b) · t1
    return (d, s, t)
```

Ví dụ 9

$\texttt{ExtEuclid}(252, 198) = (18, 4, -5)$

Gọi $\texttt{ExtEuclid}(\cdot, \cdot)$	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

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 $a \mid c$

□

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 $a \equiv b \pmod{m}$

□

- 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 *nghịch đảo (inverse)* của a theo môđun m, ký hiệu a^{-1}, là bất kỳ số nguyên nào thỏa mãn $a^{-1}a \equiv 1 \pmod{m}$
  - Đôi khi ta cũng dùng ký hiệu $\overline{a}$ thay vì a^{-1}
  - Chú ý rằng nếu ta có thể tìm được a^{-1} thỏa mãn điều kiện trên, ta có thể giải $ax \equiv b \pmod{m}$ bằng cách nhân cả hai vế với a^{-1}, nghĩa là, $a^{-1}ax \equiv a^{-1}b \pmod{m}$, suy ra $1 \cdot x \equiv a^{-1}b \pmod{m}$, và do đó $x \equiv a^{-1}b \pmod{m}$

Định lý 15

Nếu $\gcd(a, m) = 1$ *và* $m > 1$ *thì tồn tại nghịch đảo* a^{-1} *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 $a^{-1} = s$
- 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 8

Chứng minh rằng nếu $\gcd(a, m) > 1$ với a là số nguyên bất kỳ và $m > 2$ là một số nguyên dương thì không tồn tại một nghịch đảo của a theo môđun m

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 10

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 $5, -9, 12, \dots$) đều là nghịch đảo của -3 theo môđun 7

Ví dụ 11

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 $x \equiv 6 \pmod 7$
$$3x \equiv 3 \cdot 6 = 18 \equiv 4 \pmod 7$$

Bài tập 9

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) $a = 89, m = 232$
(5) $a = 101, m = 4620$

Bài tập 10

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) $89x \equiv 2 \pmod{232}$

Bài tập 11

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* $n = 2$*)*

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 $m_1, m_2, \ldots, m_n$ *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* $a_1, a_2, \ldots, a_n$. *Hệ phương trình*

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_n \pmod{m_n} \end{aligned}$$

có nghiệm duy nhất theo môđun $m = m_1 m_2 \ldots m_n$. *(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*)*

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 12

*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 (**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 11 để kết luận rằng $m \mid (x - y)$ trong đó $m = m_1 m_2 \ldots m_n$)*

Ví dụ 12

Giải hệ phương trình

$$
\begin{aligned}
x &\equiv 2 \pmod{3}\\
x &\equiv 3 \pmod{5}\\
x &\equiv 5 \pmod{7}
\end{aligned}
$$

- $m = m_1 m_2 m_3 = 3 \cdot 5 \cdot 7 = 105$
- $M_1 = m/m_1 = 35$ 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 $m_2 = 7$
- $x = \sum_{i=1}^{3} a_i y_i M_i = 2 \cdot 2 \cdot 35 + 3 \cdot 1 \cdot 21 + 5 \cdot 1 \cdot 15 = 278 \equiv 68 \pmod{105}$

Ví dụ 13 (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, $x \equiv 68 \pmod{105}$

Bài tập 13

Giải hệ phương trình sau bằng các phương pháp minh họa trong hai ví dụ trước

$$x \equiv 1 \pmod 5 \tag{4}$$
$$x \equiv 2 \pmod 6 \tag{5}$$
$$x \equiv 3 \pmod 7 \tag{6}$$

Bài tập 14

Giải hệ phương trình sau bằng các phương pháp minh họa trong hai ví dụ trước

$$x \equiv 2 \pmod 3 \tag{7}$$
$$x \equiv 1 \pmod 4 \tag{8}$$
$$x \equiv 3 \pmod 5 \tag{9}$$

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

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 15 (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 $a \equiv b \pmod m$.*

(a) *Giả sử a không chia hết cho p. Chứng minh rằng không có hai số nguyên nào trong số các số $1 \cdot a, 2 \cdot a, \ldots, (p-1) \cdot a$ là đồng dư theo môđun p*

(b) *Từ phần (a), kết luận rằng tích các số $1, 2, \ldots, p-1$ đồng dư với tích các số $a, 2a, \ldots, (p-1)a$ theo môđun p. Sử dụng điều này để chứng minh rằng $(p-1)! \equiv a^{p-1}(p-1)! \pmod p$*

(c) *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)*

Ví dụ 14 (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 16

(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* $1001 = 7 \cdot 11 \cdot 13$*)*

Thuật toán mã hóa RSA

Mật mã khóa công khai

- Trong *mật mã khóa bí mật (private key cryptography)*, một khóa bí mật được sử dụng cả trong việc mã hóa lẫn giải mã các thông điệp
  - Một vấn đề đặt ra là làm sao để *chia sẻ khóa bí mật một cách an toàn*
- Trong *mật mã khóa công khai (public key cryptography)*, hai khóa được sử dụng: một để mã hóa và một để giải mã
  - Thông tin gửi đến có thể được mã hóa bởi bất kỳ ai có khóa công khai, nhưng chỉ có thể được giải mã bởi người sở hữu khóa bí mật
  - Người sở hữu khóa bí mật có thể mã hóa thông tin với khóa bí mật của mình, và bất kỳ ai cũng có thể giải mã thông tin này bằng khóa công khai, và biết rằng chỉ có duy nhất người sở hữu khóa bí mật có thể mã hóa thông tin đó. (Đây là cơ sở của chữ ký điện tử)
- Hệ mã khóa công khai được biết đến nhiều nhất là RSA

Thuật toán mã hóa RSA

RSA - Rivest-Shamir-Adleman

- 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 $c = m^e \bmod n$ (Việc này có thể được thực hiện một cách hiệu quả. Xem bài giảng trước)
- **Giải mã:**
  - Tính $m = c^d \bmod n$
  - Chuyển m từ số nguyên sang thông điệp M ban đầu

Thuật toán mã hóa RSA
RSA - Rivest-Shamir-Adleman

Ví dụ 15

- $n = pq = 43 \cdot 59 = 2537$, $k = 42 \cdot 58 = 2436$
- 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 $M =$ 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: ST $\Rightarrow 1819$ và 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

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 $m^{ed} \equiv m \pmod{pq} \equiv m \pmod{n}$

$\square$

Tài liệu tham khảo

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.

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
Hàm sinh
Một số ví dụ khác

- *Quy tắc song ánh (The Bijection Rule):* Nếu $f : A \to B$ là một song ánh, trong đó A và B là các tập hữu hạn, thì $|A| = |B|$

Ví dụ 1

- Cho tập hợp $U = \{u_1, u_2, \dots, u_n\}$
- Mỗi tập con $A \subseteq U$ được biểu diễn bằng chuỗi nhị phân $x_1x_2 \dots x_n$ trong đó $x_i = 1$ nếu $u_i \in A$ và $x_i = 0$ nếu $u_i \notin A$
- Số tập con của U bằng với số các 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
    - $\dots$
    - 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 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

$$|A_1 \times A_2 \times \cdots \times A_n| = |A_1| \cdot |A_2| \cdot \ldots \cdot |A_n| = m_1 m_2 \ldots m_n$$

- *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

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = |A_1| + |A_2| + \cdots + |A_n| = m_1 + m_2 + \cdots + m_n$$

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)
  - $\dots$
  - 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

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
  - $\dots$
  - 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 $f : A \to B$ với A và B lần lượt là các tập hữu hạn gồm m và n phần tử?

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 $23 + 15 + 19 = 57$

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 $i = 2$, có $7 - 2 = 5$ lựa chọn cho j
  - $\dots$
  - 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 xác hai số 1 là $\sum_{i=1}^{7}(7 - i) = \sum_{i=0}^{6} i = 6(6 + 1)/2 = 21$

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

$$\begin{aligned}\text{số tên biến} &= \text{số tên biến dạng } x + \text{số tên biến dạng } x\alpha \\ &= 26 + 260 = 286\end{aligned}$$

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 này trên trang web môn học `https://hoanganhduc.github.io/teaching/VNU-HUS/2024/MAT3500/Lectures/Counting.pdf` sau khi rút gọn thông qua bit.ly là `https://bit.ly/3U4FW1V`

Giả sử các đường dẫn sau khi rút gọn gồm có `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. Có tất cả bao nhiêu đường dẫn rút gọn?

- Mỗi đường dẫn tương ứng với một chuỗi ký tự $x_1x_2\dots x_7$ trong đó $x_i \in C = \{0,\dots,9\} \cup \{A,\dots,Z\} \cup \{a,\dots,z\}$, $1 \leq i \leq 7$
  - Có $|C|$ cách chọn giá trị cho x_1
  - Với mỗi giá trị của x_1, có $|C|$ cách chọn giá trị cho x_2
  - $\dots$
  - Với mỗi giá trị của $x_1,\dots,x_6$, có $|C|$ cách chọn giá trị cho x_7
- Theo quy tắc nhân, có $|C|^7 = (10+26+26)^7 = 62^7 = 3\,521\,614\,606\,208$ đường dẫn rút gọn

Bài tập 2

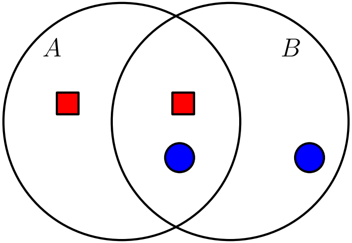
Địa chỉ kiểm soát truy cập phương tiện truyền thông (MAC (media access control) address), hay còn gọi là địa chỉ MAC, là một định danh duy nhất được gán cho một card mạng (network adapter), ví dụ như card mạng có dây (ethernet card) hoặc card mạng không dây (wireless card). Địa chỉ này gồm một dãy sáu 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?*

- *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

$$|A \cup B| = |A| + |B| - |A \cap B|$$



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 $x_1x_2 \dots x_6 00$ (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 $x_1x_2 \dots x_6 00$ là 2^6
  - $|A \cap B|$: Số chuỗi nhị phân độ dài 8 có dạng $1x_2 \dots x_6 00$ là 2^5

Số chuỗi nhị phân độ dài 8 bắt đầu với 1 hoặc kết thúc với 00 là $|A \cup B| = 2^7 + 2^6 - 2^5 = 160$

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 $xxxy$ 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 $xyyy$ 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 $xyzt$ 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 $|S \cup E| = 10^2 + 10^2 - 10 = 190$

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 $|A|$. Các chuỗi nhị phân độ dài 10 có chứa 00000 thuộc một trong các dạng: $00000x_6x_7x_8x_9x_{10}$, $100000x_7x_8x_9x_{10}$, $x_1100000x_8x_9x_{10}$, $x_1x_2100000x_9x_{10}$, $x_1x_2x_3100000x_{10}$, và $x_1x_2x_3x_4100000$ (lần lượt ứng với các vị trí bắt đầu dãy 00000)
  - 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, $|A \cup B| = 2(2^5 + 5 \cdot 2^4) - 2 = 222$

Bài tập 4

Có bao nhiêu chuỗi nhị phân độ dài 7 *bắt đầu với* 00 *hoặc kết thúc với* 111*?*

Bài tập 5

Một chuỗi đối xứng 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 6

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

Bài tập 7

Có tất cả bao nhiêu số nguyên không vượt quá 1000 là bình phương hoặc lập phương của một số nguyên dương?

Bài tập 8 ($\star$)

*Có bao nhiêu chuỗi nhị phân độ dài 8 có chứa 000 hoặc 1111? (**Đáp án:** 147)*

- *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 $f : A \to B$ 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 $n/d = 280/4 = 70$

Ví dụ 12

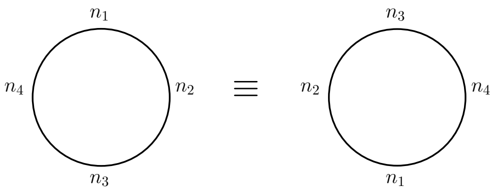
Có bao nhiêu cách khác nhau để sắp xếp các số $1, 1, 2$?

- 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 $3!/2 = 3$

$1_a, 1_b, 2$	$1_b, 1_a, 2$	$1, 1, 2$
$1_a, 2, 1_b$	$1_b, 2, 1_a$	$1, 2, 1$
$2, 1_a, 1_b$	$2, 1_b, 1_a$	$2, 1, 1$

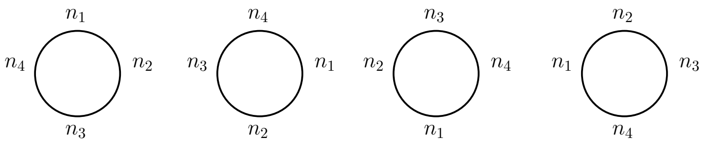
Ví dụ 13

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 nguyên lý đếm cơ bản
Quy tắc chia

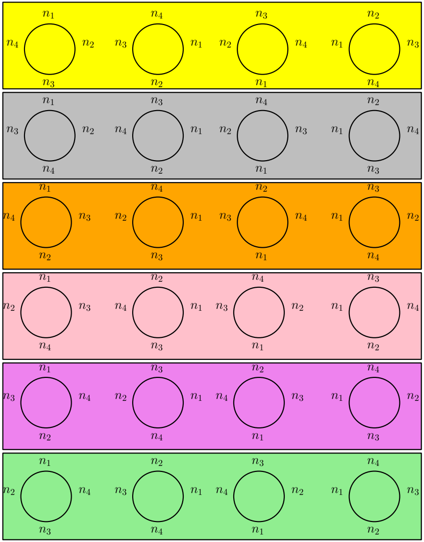
- Nếu *không có điều kiện gì* thì có tất cả $4!$ cách sắp xếp
- Chú ý rằng hai cách sắp xếp là giống nhau nếu khi ta xoay bàn sao cho n_1 nằm ở trên đỉnh thì chúng giống nhau
- Do đó, *với mỗi cách sắp xếp 4 người quanh bàn tròn, có chính xác 4 trong $4!$ cách sắp xếp "cùng loại"*
- Theo quy tắc chia, số cách khác nhau để xếp 4 người quanh bàn tròn là $4!/4 = 6$



Hình: Bốn cách sắp xếp giống nhau

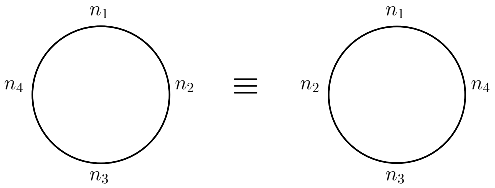
Các nguyên lý đếm cơ bản

Quy tắc chia



Bài tập 9

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?*

Bài tập 10

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?*

Nguyên lý chuồng bồ câu

- *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 $f : A \to B$ 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, 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

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"?
  Trong 145 sinh viên (= "bồ câu"), có ít nhất $\lceil 145/12 \rceil = 13$ sinh viên sinh ra vào cùng một tháng (= "chuồng"). Thêm vào đó, *phát biểu với $d \geq 14$ không đúng* khi 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 đề bài

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 $1 \leq i \leq n+1$
- Có tối đa n số nguyên dương lẻ (= "chuồng") nhỏ hơn $2n$. 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 $a_j \mid a_i$

Nguyên lý chuồng bồ câu

Ví dụ 18

Mọi dãy số gồm n^2+1 số thực phân biệt có một dãy con gồm $n+1$ phần tử và là dãy thực sự tăng hoặc thực sự giảm

- Gọi dãy $a_1, a_2, \dots, a_{n^2+1}$ là một dãy gồm n^2+1 số thực phân biệt. Với mỗi $k \in \{1, 2, \dots, n^2+1\}$, gọi (i_k, d_k) là một cặp số tương ứng với a_k (= "bồ câu"), trong đó i_k là độ dài của dãy thực sự tăng dài nhất bắt đầu từ a_k, và d_k là độ dài của dãy thực sự giảm dài nhất bắt đầu từ a_k
- Giả sử không có dãy thực sự tăng hoặc thực sự giảm nào có $n+1$ phần tử, nghĩa là, $1 \leq i_k, d_k \leq n$. Do đó, có tối đa n^2 cặp (i_k, d_k) phân biệt (= "chuồng")
- Theo nguyên lý chuồng bồ câu, tồn tại $s, t \in \{1, 2, \dots, n^2+1\}$ thỏa mãn $(i_s, d_s) = (i_t, d_t)$. Không mất tính tổng quát, giả sử $s < t$
- Do các phần tử trong dãy đều phân biệt, ta có $a_s < a_t$ hoặc $a_t < a_s$
  - Nếu $a_s < a_t$, ta có thể xây dựng một dãy con thực sự tăng bắt đầu từ a_s gồm $i_s + 1 = i_t + 1$ phần tử bằng cách lấy a_s và dãy thực sự tăng bắt đầu từ a_t có i_t phần tử. Điều này mâu thuẫn với định nghĩa của i_s
  - Tương tự với $a_s > a_t$

Nguyên lý chuồng bồ câu

Bài tập 11

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 bao nhiêu chiếc tất để chắc chắn có hai chiếc cùng màu? bốn chiếc cùng màu?*

(b) *Cần lấy bao nhiêu chiếc tất để chắc chắn có một đôi màu đen?*

(c) *Nếu có thêm 12 chiếc tất màu nâu nữa trong ngăn tủ thì cần lấy bao nhiêu chiếc tất để chắc chắn có hai chiếc cùng màu?*

Bài tập 12

Chứng minh rằng trong một nhóm n số nguyên bất kỳ, có hai số nguyên có cùng số dư khi chia cho $n-1$

Bài tập 13 ($\star$)

*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?)*

Nguyên lý chuồng bồ câu

Bài tập 14

(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 15 ($\star$)

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 16

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*

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 r (r-permutation)* của một tập S là một dãy sắp thứ tự r phần tử phân biệt của S, trong đó r là một số nguyên thỏa mãn $0 \leq r \leq |S|$
  - Tập S có tất cả sáu chỉnh hợp chập 2: $(1, 2)$, $(2, 1)$, $(1, 3)$, $(3, 1)$, $(2, 3)$, $(3, 2)$
- Ký hiệu $P(n, r)$ hoặc P_n^r là *số chỉnh hợp chập r của (một tập) n phần tử*

Định lý 1

Với mọi số nguyên $n \geq 1$ và mọi số nguyên r thỏa mãn $0 \leq r \leq n$

$$P_n^r = \frac{n!}{(n-r)!}$$

Chứng minh Định lý 1.

Để xây dựng một chỉnh hợp chập r của một tập S gồm n phần tử, ta thực hiện một dãy r bước chọn các phần tử phân biệt trong S để xếp vào r 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
- $\dots$
- Với $r-1$ vị trí đầu đã được xác định, chọn phần tử xếp vào vị trí thứ r: có $n-r+1$ cách chọn

Theo quy tắc nhân, có tất cả

$n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot (n-r+1) = \dfrac{n!}{(n-r)!}$ chỉnh hợp chập r

của một tập n phần tử. $\square$

Ví dụ 19

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 20

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 $7! = 5040$

Ví dụ 21

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 $1 \cdot 3! = 6$

- Một *tổ hợp chập r (r-combination)* của một tập hợp S là một dãy không sắp thứ tự r phần tử phân biệt của S, trong đó r là một số nguyên thỏa mãn $0 \leq r \leq |S|$. Nói cách khác, một tổ hợp chập r của S cũng là một tập con r phần tử của S
  - Tập $S = \{1, 2, 3\}$ có ba tổ hợp chập 2: $\{1, 2\}$, $\{2, 3\}$, $\{1, 3\}$
- Ký hiệu $C(n, r)$, C_n^r, hoặc $\binom{n}{r}$ (đọc là "n chọn r") là *số tổ hợp chập r của (một tập) n phần tử*

Định lý 2

Với mọi số nguyên $n \geq 1$ và mọi số nguyên r thỏa mãn $0 \leq r \leq n$

$$C_n^r = \frac{n!}{r!(n-r)!}$$

Chứng minh Định lý 2.

Để xây dựng một tổ hợp chập r 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 r của S: có P_n^r cách
- Theo định nghĩa, trong mỗi tổ hợp chập r của S, thứ tự giữa các phần tử là không quan trọng. Nói cách khác, mỗi tổ hợp chập r của S ứng với chính xác $P_r^r = r!$ chỉnh hợp chập r 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 r của S là

$$C_n^r = \frac{P_n^r}{P_r^r} = \frac{n!}{r!(n-r)!}$$

□

Ví dụ 22

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 $C_{52}^{7} = \dfrac{52!}{7!45!} = 133\,784\,560$

Ví dụ 23

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

Định lý 3

Với các số nguyên n, r thỏa mãn $0 \leq r \leq n$, ta có

$$C_n^r = C_n^{n-r}$$

Chứng minh.

- Giả sử $\mathcal{A}$ là tập các tập con r phần tử và $\mathcal{B}$ là tập các tập con $n-r$ phần tử của một tập n phần tử X. Hàm $f : \mathcal{A} \to \mathcal{B}$ cho bởi $f(A) = X \setminus A$ với $A \in \mathcal{A}$ là một song ánh.
- Do đó, $C_n^r = C_n^{n-r}$

□

Bài tập 17

Chứng minh Định lý 3 bằng công thức đã biết.

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 tiếp theo, với $n, k \geq 2$ và $n \geq k + 2$, ta minh họa phương pháp nêu trên bằng cách chứng minh

$$C_n^k - C_{n-2}^{k-2} = 2C_{n-2}^{k-1} + C_{n-2}^k$$

Ví dụ 24 (Đếm bằng hai cách)

Giả sử $n, k \geq 2$ và $n \geq k+2$. Có bao nhiêu cách chọn k số nguyên từ tập $\{1, 2, \ldots, n\}$ 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 $\{3, \ldots, n\}$
  - 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 $\{3, \ldots, n\}$
  - 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 $\{3, \ldots, n\}$
  - Do đó đáp án là $2C_{n-2}^{k-1} + C_{n-2}^{k}$

Ta đã chứng minh $C_n^k - C_{n-2}^{k-2} = 2C_{n-2}^{k-1} + C_{n-2}^{k}$

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, $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$

□

Bài tập 18

Chứng minh Định lý 4 bằng công thức đã biết

Bài tập 19

Sử dụng Định lý 4, chứng minh đẳng thức thu được từ Ví dụ 24 sau

$$C_n^k - C_{n-2}^{k-2} = 2C_{n-2}^{k-1} + C_{n-2}^k$$

Tam giác Pascal và Định lý nhị thức
Tam giác Pascal

Bắt đầu với $n = 0$, hàng thứ n có $n + 1$ phần tử: $C_n^0, C_n^1, \ldots, C_n^n$

$$C_0^0 = 1$$

$$C_1^0 = 1 \qquad C_1^1 = 1$$

$$C_2^0 = 1 \qquad C_2^1 = 2 \qquad C_2^2 = 1$$

$$C_3^0 = 1 \qquad C_3^1 = 3 \qquad C_3^2 = 3 \qquad C_3^3 = 1$$

$$C_4^0 = 1 \qquad C_4^1 = 4 \qquad C_4^2 = 6 \qquad C_4^3 = 4 \qquad C_4^4 = 1$$

Hình: Tam giác Pascal

Định lý 3 và 4 có thể được minh họa bằng tam giác Pascal

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 20

*Chứng minh Định lý 5 bằng phương pháp quy nạp. (**Gợi ý:** Sử dụng Định lý 4)*

Bài toán: Khai triển $(x+y)^n$

Ví dụ 25

$$
\begin{aligned}
(x+y)^0 &= 1 \\
(x+y)^1 &= 1 \cdot x + 1 \cdot y \\
(x+y)^2 &= 1 \cdot x^2 + 2 \cdot xy + 1 \cdot y^2 \\
(x+y)^3 &= 1 \cdot x^3 + 3 \cdot x^2y + 3 \cdot xy^2 + 1 \cdot y^3 \\
(x+y)^4 &= 1 \cdot x^4 + 4 \cdot x^3y + 6 \cdot x^2y^2 + 4 \cdot xy^3 + 1 \cdot y^4
\end{aligned}
$$

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

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 21

Chứng minh Định lý 5 bằng cách sử dụng Định lý 6

Ví dụ 26

Khai triển $(x+2)^4$

$$(x+2)^4 = C_4^0 \cdot x^4 + C_4^2 \cdot x^3 \cdot 2^1 + C_4^3 \cdot x^2 \cdot 2^2 + \\ + C_4^1 \cdot x^1 \cdot 2^3 + C_4^4 \cdot x^0 \cdot y^4 \\ = x^4 + 8x^3 + 24x^2 + 32x + 16$$

Ví dụ 27

Tính $(1.02)^7$ làm tròn đến chữ số thập phân thứ 4

$$(1+0.02)^7 = C_7^0 \cdot 1^7 + C_7^1 \cdot 1^6 \cdot (0.02) + C_7^2 \cdot 1^5 \cdot (0.0004) \\ + C_7^3 \cdot 1^4 \cdot (0.000008) + \dots \\ = 1 + 1.14 + 0.0084 + 0.00028 + \dots \\ \approx 1.14868 \approx 1.1487$$

Chú ý: Để làm tròn đến chữ số thập phân thứ 4, ta cần tính toán đến chữ số thập phân thứ 5

Bài tập 22

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) *($\star$) 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 23

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) *($\star$) của x^k trong khai triển của $\left(x+\frac{1}{x}\right)^{100}$, trong đó k là một số nguyên*

Định lý 7: Nguyên lý bù trừ tổng quát

Với các tập hữu hạn $A_1, A_2, \dots, A_n$

$$\left|\bigcup_{k=1}^{n} A_k\right| = \sum_{k=1}^{n}\left(\sum_{\{I \mid I \subseteq \{1,\dots,n\} \wedge |I|=k\}} (-1)^{k-1}\left|\bigcap_{i \in I} A_i\right|\right)$$

- $n = 2$

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

- $n = 3$

$$\begin{aligned}|A_1 \cup A_2 \cup A_3| = |A_1| + |A_2| + |A_3| \\ - |A_1 \cap A_2| - |A_2 \cap A_3| - |A_1 \cap A_3| \\ + |A_1 \cap A_2 \cap A_3|\end{aligned}$$

Chứng minh.

$$\left|\bigcup_{k=1}^{n} A_k\right| = \sum_{k=1}^{n}\left(\sum_{\{I \mid I\subseteq\{1,\dots,n\}\wedge|I|=k\}} (-1)^{k-1}\left|\bigcap_{i\in I} A_i\right|\right)$$

- 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 vế trái
- Có bao nhiêu lần a được đếm ở vế phải?
  - a xuất hiện trong m 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 $1 \leq i < j < k \leq m$
  - $\dots$

  Do đó, ở vế phải, a được đếm $\sum_{k=1}^{m}(-1)^{k-1}C_m^k$ lần

□

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

$$\begin{aligned} 0 &= (-1+1)^m \\ &= \sum_{k=0}^{m} C_m^k(-1)^k 1^{m-k} \\ &= 1 + \sum_{k=1}^{m}(-1)^k C_m^k \end{aligned}$$

Do đó, $\sum_{k=1}^{m}(-1)^k C_m^k = -1$. Suy ra $\sum_{k=1}^{m}(-1)^{k-1}C_m^k = 1$

(Chia hai vế cho -1)

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* $m > 0$, $\sum_{k=1}^{m} (-1)^{k-1} C_m^k = 1$. *(**Gợi ý:** $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$)*

Bài tập 25

Chứng minh đẳng thức Vandermonde (Vandermonde's Identity) sau

$$C_{m+n}^r = \sum_{k=0}^{r} C_m^{r-k} C_n^k,$$

trong đó m, n, r *là các số nguyên không âm và* $r \leq \min(m, n)$. *(**Gợi ý:** Có bao nhiêu cách chọn ra một ban đại diện gồm* r *thành viên từ một lớp học có* m *nam và* n *nữ?)*

Chỉnh hợp và tổ hợp suy rộng

- Một *chỉnh hợp lặp chập r (r-permutation with repetition)* của một tập S là một dãy sắp thứ tự r phần tử của S trong đó các phần tử có thể lặp lại
- Một *tổ hợp lặp chập r của một tập S (r-combination with repetition)* của một tập S là một dãy không sắp thứ tự r phần tử của S trong đó các phần tử có thể lặp lại

Định lý 8

Số chỉnh hợp lặp chập r của một tập n phần tử là n^r

Chứng minh.

Có n lựa chọn cho vị trí thứ nhất, n lựa chọn cho vị trí thứ hai, ..., n lựa chọn cho vị trí thứ r. Do đó có n^r cách xây dựng một chỉnh hợp lặp chập r của một tập n phần tử □

Chỉnh hợp và tổ hợp suy rộng

Định lý 9

Số tổ hợp lặp chập r của một tập n phần tử là C^{n-1}_{n+r-1}

Chứng minh.

- Mỗi tổ hợp lặp chập r từ tập n phần tử có thể biểu diễn bằng một dãy $n-1$ thanh đứng và r 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
- Mỗi dãy $n-1$ thanh đứng và r ngôi sao ứng với một chuỗi nhị phân độ dài $n+r-1$ có chính xác r số 1. Do đó, số các dãy thanh đứng và ngôi sao là $C^{r}_{n+r-1} = C^{n-1}_{n+r-1}$

$\square$

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 $** \mid *** \mid *$
  - Lựa chọn [R, B, R, R, B, R] ứng với dãy $* * ** \mid\mid **$
- Số cách để 6 quả bóng vào túi là $C^{3-1}_{3+6-1} = 28$

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: $C^{3-1}_{3+11-1} = 78$

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 $C^{3-1}_{3+12-1} = 91$

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 $C^{11}_{3+11-1} = 78$ nghiệm nguyên thỏa mãn $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$. Có $C^{8}_{3+8-1} = 45$ nghiệm nguyên thỏa mãn $x_1 \geq 3$, $x_2 \geq 0$, $x_3 \geq 0$. Do đó, kết quả là $C^{11}_{3+11-1} - C^{8}_{3+8-1} = 78 - 45 = 33$

Bài tập 26

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* $x_1 = 2$

Định lý 10

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, $\ldots$, và n_k phần tử giống nhau thuộc loại k, là $\dfrac{n!}{n_1!n_2!\ldots n_k!}$

Chứng minh.

- Nếu coi tất cả n phần tử đều khác nhau, có $n!$ hoán vị
- Với mỗi hoán vị trong $n!$ hoán vị này, có thể xây dựng một hoán vị giống nó bằng một chuỗi k bước:
  - Hoán vị n_1 phần tử loại 1: có $n_1!$ cách
  - Với mỗi cách hoán vị các phần tử loại 1, hoán vị n_2 phần tử loại 2: có $n_2!$ cách
  - ...
  - Với mỗi cách hoán vị các phần tử loại $1, \ldots, k-1$, hoán vị n_k phần tử loại k: có $n_k!$ cách

Theo quy tắc nhân, mỗi hoán vị có $n_1!n_2!\ldots n_k!$ hoán vị giống nó trong số $n!$ hoán vị

Chỉnh hợp và tổ hợp suy rộng

Ví dụ 32

Có bao nhiêu chuỗi ký tự 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 kết quả là $\dfrac{7!}{3!2!1!1!} = 420$

Chỉnh hợp và tổ hợp suy rộng

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 chuỗi các bước
  - Chia 5 quân từ 52 quân bài cho người chơi thứ nhất: có C^5_{52} cách
  - Với mỗi bộ 5 quân mà người chơi thứ nhất có, chia tiếp 5 quân từ 47 quân còn lại cho người chơi thứ hai: có C^2_{47} cách
  - Với mỗi các bộ 5 quân mà hai người chơi đầu tiên có, chia tiếp 5 quân từ 42 quân còn lại cho người chơi thứ ba: có C^5_{42} cách
  - Với mỗi các bộ 5 quân mà ba người chơi đầu tiên có, chia tiếp 5 quân từ 37 quân còn lại cho người chơi thứ tư: có C^5_{37} cách
- Theo quy tắc nhân, có tất cả $C^5_{52}C^5_{47}C^5_{42}C^5_{37} = \dfrac{52!}{5!5!5!5!32!}$ cách chia bài

Chỉnh hợp và tổ hợp suy rộng

Định lý 11

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

$$\frac{n!}{n_1! n_2! \dots n_k!}$$

Với các số nguyên $n, n_1, n_2, \dots, n_k \geq 0$ 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 12: Định lý đa thức

Với mọi $n \geq 0$ và $k \geq 1$

$$(x_1 + \dots + x_k)^n = \sum_{\substack{0 \leq n_1, \dots, n_k \leq n \\ n_1 + \dots + n_k = n}} C_n^{n_1, n_2, \dots, n_k} x_1^{n_1} \dots x_k^{n_k}$$

Hàm sinh

Hàm sinh

Hàm sinh (generating function) $G_a(x)$ của một dãy vô hạn $\{a_n\}$ $(n \geq 0)$ được định nghĩa bởi $G_a(x) = \sum_{n=0}^{\infty} a_n x^n$. Nói cách khác, a_n là hệ số của x^n trong $G_a(x)$

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*

Ví dụ 34 (Định lý nhị thức)

- Hàm sinh của dãy $C_n^0, C_n^1, \dots, C_n^n, 0, 0, \dots$ là

$$G(x) = \sum_{k=0}^{n} C_n^k x^k = (1+x)^n$$

- *Hệ số của x^k* trong $G(x)$ là *số cách chọn ra k số x và $n-k$ số 1 từ n thừa số của tích*

$$(1+x)^n = (1+x)(1+x)\dots(1+x),$$

trong đó với mỗi thừa số ta chọn chính xác một phần tử: hoặc 1 hoặc x

  - Có C_n^k cách chọn k số x
  - Ứng với mỗi cách chọn k số x, có chính xác 1 cách chọn $n-k$ số 1 từ $n-k$ thừa số $(1+x)$ còn lại

Ví dụ 35

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*
- Một phương án là $G(x) = (1 + x + x^2 + \cdots + x^{11})^3$
  - Để có x^{11}, ta cần chọn lần lượt x^{x_1}, x^{x_2}, và x^{x_3} từ ba thừa số $(1 + x + \cdots + x^{11})$ của $G(x)$ (từ mỗi thừa số, chọn chính xác một số hạng trong tổng $1 + x + \cdots + x^{11}$) thỏa mãn $x_1 + x_2 + x_3 = 11$
  - Số cách chọn thỏa mãn điều kiện trên chính là hệ số của x^{11} và cũng là số nghiệm nguyên không âm của $x_1 + x_2 + x_3 = 11$
- Ta có

$$
\begin{aligned}
G(x) &= (1 + x + \cdots + x^{11})^3 \\
&= \left(\frac{1 - x^{12}}{1 - x}\right)^3 \\
&= (1 - x^{12})^3 (1 - x)^{-3}
\end{aligned}
$$

Hàm sinh

Để tìm hệ số của x^{11} trong Ví dụ 35, ta cần một số định nghĩa và định lý sau

Hệ số nhị thức mở rộng

Với $n \in \mathbb{R}$ bất kỳ và $r \geq 0$, *hệ số nhị thức tổng quát (generalized binomial coefficient)* C_n^r được định nghĩa như sau

$$C_n^r = \frac{n(n-1)\dots(n-r+1)}{r!}$$

Ví dụ 36

$$C_{-2}^5 = \frac{(-2)(-3)(-4)(-5)(-6)}{5!} = -6$$

Hàm sinh

Mệnh đề 13

Nếu n là một số nguyên dương thì

$$C^r_{-n} = (-1)^r C^r_{n+r-1}$$

Định lý 14: Định lý nhị thức tổng quát

Với mọi $n \in \mathbb{R}$,

$$(1+x)^n = \sum_{r=0}^{\infty} C^r_n x^r$$

Ví dụ 37

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

$$(-1)^r C^r_{-3} = (-1^r)\big[(-1)^r \cdot C^r_{3+r-1}\big] = \frac{(r+2)(r+1)}{2}$$

- Suy ra, hệ số của x^{11} trong khai triển của $(1-x)^{-3}$ là $(11+2)(11+1)/2 = 78$

Ví dụ 38 (Tiếp tục Ví dụ 35)

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 37)

Ví dụ 39

Phương trình $x_1 + x_2 + x_3 = 6$ có bao nhiêu nghiệm nguyên thỏa mãn $-1 \leq x_1 \leq 2$ và $1 \leq x_2, x_3 \leq 4$

- Ta định nghĩa

$$G(x) = (x^{-1} + x^0 + x^1 + x^2)(x^1 + x^2 + x^3 + x^4)^2$$

- 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 27

Hoàn thành Ví dụ 39 bằng cách tìm hệ số của x^5 trong khai triển của $(1 + x + x^2 + x^3)^3$

Bài tập 28

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* $\dfrac{1+x}{(1-2x)^5}$

*(**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 29

Sử dụng hàm sinh để đếm số nghiệm nguyên của $x_1 + x_2 + x_3 = 11$ *thỏa mãn* $0 \le x_1 \le 2$, $x_2 \ge 0$, *và* $x_3 \ge 0$

Bài tập 30

Sử dụng hàm sinh để đếm số nghiệm nguyên của $x_1 + x_2 + x_3 = 11$ *thỏa mãn* $x_1 \ge -2$, $x_2 \ge 1$, *và* $x_3 \ge 0$. *(**Gợi ý:** Chú ý rằng* $-2 \le x_1 \le 11$, $1 \le x_2 \le 13$, *và* $0 \le x_3 \le 12$*)*

Một số ví dụ khác

Ví dụ 40

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

Một số ví dụ khác

Ví dụ 41

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
  - $5, 1$
  - $4, 2 \quad 4, 1, 1$
  - $3, 3 \quad 3, 2, 1 \quad 3, 1, 1, 1$
  - $2, 2, 2 \quad 2, 2, 1, 1$

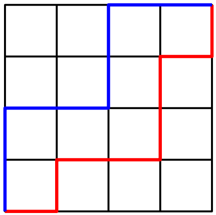
Một số ví dụ khác

Bài tập 31 ($\star$)

*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 32

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

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

- 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

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

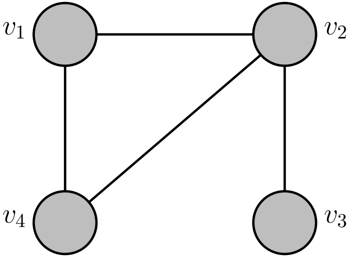
Một số ví dụ

- Một *đồ thị (graph)* G bao gồm một tập các *đỉnh (vertex)* hoặc *nút (node)* V và một tập cách cạnh E nối các (cặp) đỉnh với nhau
- Có nhiều loại đồ thị khác nhau (vô hướng, có hướng, đồ thị đơn giản, đa đồ thị, v.v...), mỗi loại có cách định nghĩa cụ thể khác nhau, tùy thuộc vào việc các loại cạnh nào cần được xét
- Điều này dẫn tới việc tồn tại nhiều thuật ngữ khác nhau (và thường không thống nhất)
- Trước khi đi vào định nghĩa đồ thị một cách cụ thể, chúng ta xét một số ví dụ

Ví dụ 1 (Đơn đồ thị vô hướng (simple undirected graph))

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_4, v_2v_3, v_2v_4\}$$

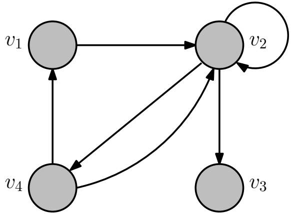


Hình: Chỉ có các cạnh *vô hướng*; có *nhiều nhất một cạnh* nối hai đỉnh phân biệt bất kỳ; và không có *khuyên (loop)*—cạnh nối giữa một đỉnh và chính nó

Ví dụ 2 (Đồ thị có hướng (và có khuyên) (directed graph (with loops)))

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{(v_1, v_2), (v_2, v_2), (v_2, v_3), (v_2, v_4), (v_4, v_1), (v_4, v_2)\}$$

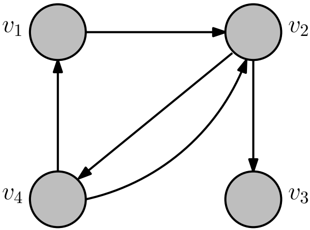


Hình: Chỉ có các cạnh *có hướng*; có *nhiều nhất một cạnh có hướng* nối từ một đỉnh bất kỳ sang một đỉnh khác bất kỳ; và *có khuyên*

Ví dụ 3 (Đơn đồ thị có hướng (simple directed graph))

$$V = \{v_1, v_2, v_3, v_4\}$$

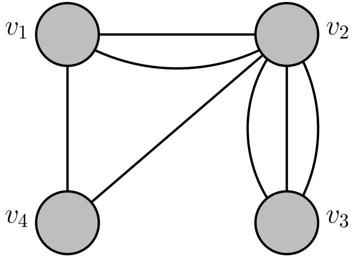
$$E = \{(v_1, v_2), (v_2, v_3), (v_2, v_4), (v_4, v_1), (v_4, v_2)\}$$



Hình: Chỉ có các cạnh *có hướng*; có *nhiều nhất một cạnh có hướng* nối từ một đỉnh bất kỳ sang một đỉnh khác bất kỳ; và *không có khuyên*

Ví dụ 4 (Đa đồ thị vô hướng (undirected multigraph))

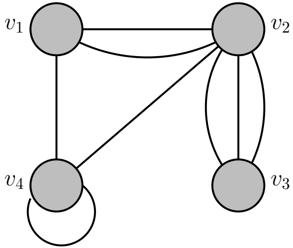
$$V = \{v_1, v_2, v_3, v_4\}$$
$$E = \{v_1v_2, v_1v_4, v_2v_3, v_2v_4\}$$
$$m(v_1v_2) = 2,\ m(v_2v_3) = 3$$
$$m(v_1v_4) = m(v_2v_4) = 1$$



Hình: Chỉ có các cạnh *vô hướng*; có thể có *nhiều cạnh* nối giữa hai đỉnh bất kỳ; và *không có khuyên*

Ví dụ 5 (Đa đồ thị vô hướng có khuyên (undirected pseudograph))

$$V = \{v_1, v_2, v_3, v_4\}$$
$$E = \{v_1v_2, v_1v_4, v_2v_3, v_2v_4, v_4v_4\}$$
$$m(v_1v_2) = 2,\ m(v_2v_3) = 3$$
$$m(v_1v_4) = m(v_2v_4) = m(v_4, v_4) = 1$$

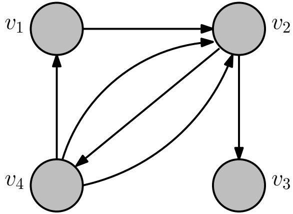


Hình: Chỉ có các cạnh *vô hướng*; có thể có *nhiều cạnh* nối giữa hai đỉnh bất kỳ; và *có khuyên* (có thể có nhiều khuyên tại một đỉnh)

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)

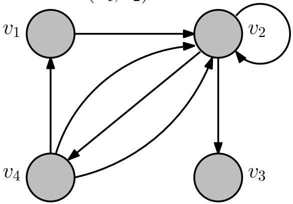
Ví dụ 7 (Đa đồ thị có hướng và có khuyên (directed pseudograph))

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{(v_1, v_2), (v_2, v_2), (v_2, v_3), (v_2, v_4), (v_4, v_1), (v_4, v_2)\}$$

$$m(v_1, v_2) = m(v_2, v_2) = m(v_2, v_3) = m(v_2, v_4) = m(v_4, v_1) = 1$$

$$m(v_4, v_2) = 2$$



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
Một số ví dụ

	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	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	*Không*[1]
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 sách của Rosen
- Các đồ thị sẽ được đề cập trong bài giảng
  - đơn đồ thị vô hướng ((simple, undirected) graph)
  - đồ thị có hướng (directed graph hoặc digraph)

[1] Khác với sách của Rosen

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

- Một đồ thị có hướng $G = (V, E)$ đơn giản là một tập hợp V cùng với một *quan hệ nhị phân (binary relation)* E trên V

- 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 *đơn đồ thị vô hướng (simple, 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 cách *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 mút (endpoint)* của e. Ta nói các đỉnh u, v là *liền kề (adjacent)* trong đồ thị G, và cạnh e gọi là cạnh *liên thuộc (incident)* với các đỉnh u, v

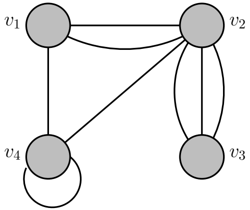
- Định nghĩa trên có thể áp dụng cho cả trường hợp V là tập có vô hạn phần tử (và đồ thị tương ứng được gọi là *đồ thị vô hạn (infinite graph)*). Tuy nhiên, trong bài giảng, chúng ta chỉ đề cập đến các *đồ thị hữu hạn (finite graph)*.

Cho $G = (V, E)$ là một đồ thị vô hướng

- Tập hợp các đỉnh kề với đỉnh v của G, ký hiệu $N(v)$ hay $N_G(v)$, được gọi là *tập láng giềng (neighborhood)* của v.
- Với một tập các đỉnh $A \subseteq V$, ta ký hiệu $N(A)$ hoặc $N_G(A)$ để chỉ tập các đỉnh liền 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

- $N(v_1) = \{v_2, v_4\}$, $N(v_2) = \{v_1, v_3, v_4\}$, $N(v_3) = \{v_2\}$, $N(v_4) = \{v_1, v_2, v_4\}$
- $\deg(v_1) = \deg(v_3) = 3$, $\deg(v_2) = 6$, $\deg(v_4) = 4$



- 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)*

Định lý 1: Định lý bắt tay (Handshaking Lemma)

Cho $G = (V, E)$ là một đồ thị vô hướng có m cạnh. Ta có

$$2m = \sum_{v \in V} \deg(v)$$

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

$\square$

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 $G = (V, E)$ có m cạnh
- Ta có

$$2m = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v)$$

  - $\sum_{v \in V_1} \deg(v)$ 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

Ví dụ 9

Có bao nhiêu cạnh trong một đồ thị vô hướng có 10 đỉnh, mỗi đỉnh có bậc 6?

- Tổng bậc của các đỉnh trong đồ thị là $6 \cdot 10 = 60$
- Theo Định lý bắt tay, nếu m là số cạnh của đồ thị thì $2m = 60$, và do đó $m = 30$

Ví dụ 10

Nếu một đồ thị vô hướng có 5 đỉnh thì liệu mỗi đỉnh có thể có bậc 3 hay không?

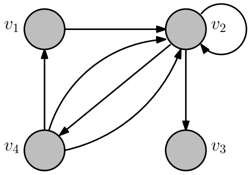
- Không. Vì nếu mỗi đỉnh có bậc 3 thì tổng bậc của các đỉnh là $3 \cdot 5 = 15$. Điều này mâu thuẫn với Định lý bắt tay: tổng bậc của các đỉnh phải là một số chẵn

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

- $\deg^-(v_1) = \deg^-(v_3) = \deg^-(v_4) = 1$, $\deg^-(v_2) = 4$
- $\deg^+(v_1) = 1$, $\deg^+(v_2) = \deg^+(v_4) = 3$, $\deg^+(v_3) = 0$



Định lý 3

Cho $G = (V, E)$ là một đồ thị có hướng. Ta có

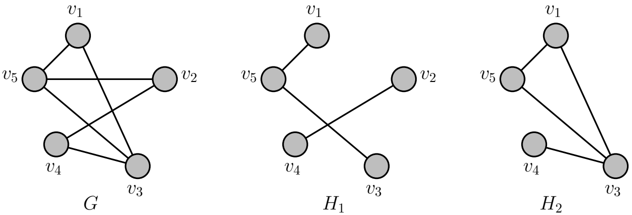
$$|E| = \sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v)$$

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, $|E|$ = tổng các bậc vào = tổng các bậc ra

$\square$

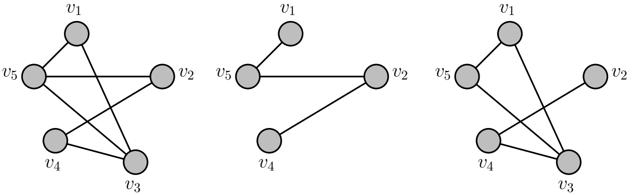
- Một *đồ thị con (subgraph)* của một đồ thị $G = (V, E)$ là một đồ thị $H = (W, F)$ trong đó $W \subseteq V$ và $F \subseteq E$
- $H = (W, F)$ là một *đồ thị con thực sự (proper subgraph)* của $G = (V, E)$ nếu H là đồ thị con của G và $H \neq G$
- $H = (W, F)$ là một *đồ thị con cảm sinh (induced subgraph)* của $G = (V, E)$ nếu H là đồ thị con của 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 $H = G[W]$



Hình: H_1 là đồ thị con thực sự của G nhưng không phải đồ thị con cảm sinh. H_2 là đồ thị con cảm sinh của G

Cho đơn đồ thị $G = (V, E)$ vô hướng và các tập $V' \subseteq V$ $E' \subseteq E$

- 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 $G - \{e\}$

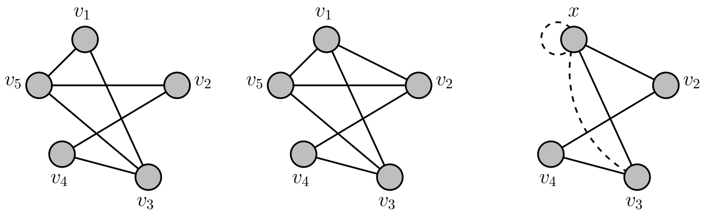


G $\qquad G - v_3$ $\qquad G - v_2v_5$

Giới thiệu
Đồ thị mới từ đồ thị cũ

Cho đơn đồ thị $G = (V, E)$ vô hướng với tập $E' \subseteq [V]^2 - E$

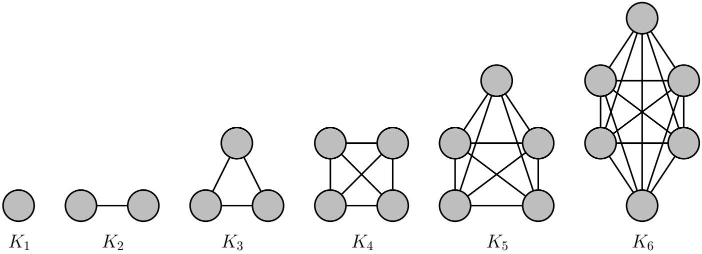
- Đồ thị $G + E'$ là đồ thị thu được bằng cách *thêm các cạnh trong E'*. Với $f \in E'$, ta viết $G + f$ thay vì $G + \{f\}$
- Đồ thị G/e là đồ thị thu được bằng *phép co (contraction) cạnh $e = uv \in E$*
  - gộp hai đỉnh u, v thành một đỉnh mới x, các cạnh kề với u và kề với v chuyển thành cạnh kề với x
  - xóa các khuyên tạo thành sau phép gộp
  - giữ lại một cạnh duy nhất trong số các cạnh song song



G $\qquad$ $G + v_1v_2$ $\qquad$ G/v_1v_5

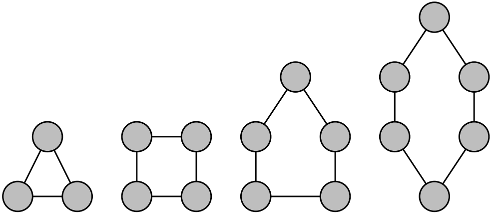
Đồ thị đầy đủ

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



Chu trình

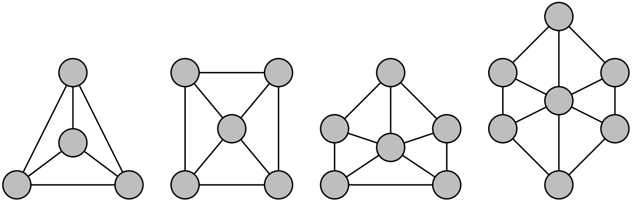
Một *chu trình (cycle)* n đỉnh với $n \geq 3$, ký hiệu C_n, là một đồ thị với các đỉnh $v_1, v_2, \dots, v_n$ và các cạnh $v_1v_2, v_2v_3, \dots, v_{n-1}v_n$, và v_nv_1



C_3 C_4 C_5 C_6

Đồ thị bánh xe

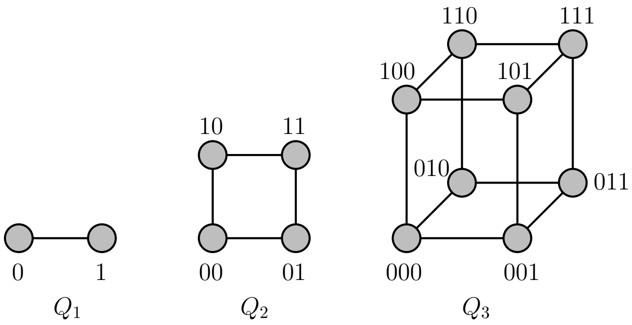
Một *đồ thị bánh xe (wheel)* gồm $n+1$ đỉnh với $n \geq 3$, ký hiệu W_n, là một đồ thị thu được bằng cách thêm một đỉnh mới vào C_n và nối đỉnh đó với mọi đỉnh của C_n bằng các cạnh mới



W_3 W_4 W_5 W_6

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 xâu nhị phân biểu diễn chúng khác nhau đúng một bit

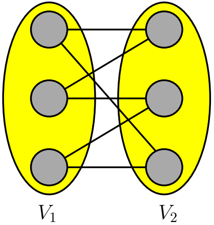


Đồ thị hai phần

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 $V = V_1 \cup V_2$, $V_1 \neq \emptyset$, $V_2 \neq \emptyset$, $V_1 \cap V_2 = \emptyset$, và mỗi cạnh của G nối một đỉnh thuộc V_1 và một đỉnh thuộc V_2. Ta cũng ký hiệu $G = (V_1 \cup V_2, E)$

Ví dụ 12

C_6 là một đồ thị hai phần



Bài tập 1

*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 2

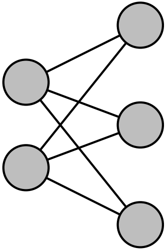
Cho đơn đồ thị vô hướng $G = (V, E)$ có $n \geq 3$ đỉnh. Gọi $H = (W, F)$ là một đồ thị con của G có ít nhất hai đỉnh. Chứng minh rằng nếu G là đồ thị hai phần thì H cũng là đồ thị hai phần.

Bài tập 3

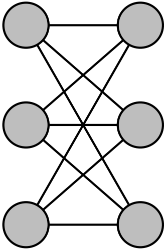
*Chứng minh W_n không là đồ thị hai phần với mọi $n \geq 3$. (**Gợi ý:** Sử dụng Bài tập 2 và kết quả K_3 không là đồ thị hai phần từ Bài tập 1)*

Đồ thị hai phần đầy đủ

Một *đồ thị hai phần đầy đủ (complete bipartite graph)* là một đồ thị hai phần $G = (V_1 \cup V_2, E)$ thỏa mãn điều kiện với mọi $v_1 \in V_1$ và $v_2 \in V_2$ ta có $v_1v_2 \in E$. Nếu $|V_1| = m$ và $|V_2| = n$, ta ký hiệu đồ thị G bằng $K_{m,n}$.



$K_{2,3}$



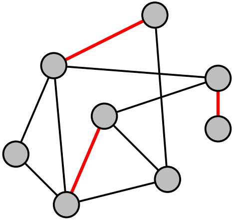
$K_{3,3}$

Giới thiệu

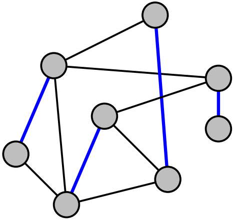
Đồ thị hai phần

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

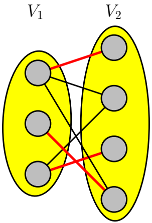


M là một ghép cặp cực đại

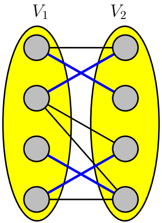
Giới thiệu
Đồ thị hai phần

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 $G = (V_1 \cup V_2, E)$, 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 $M^\star \subseteq E$ 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

Định lý 4: Định lý Hall (Hall's Marriage Theorem)

Cho $G = (V_1 \cup V_2, E)$ là một đồ thị hai phần. Tồn tại một ghép cặp $M \subseteq E$ bao phủ V_1 khi và chỉ khi với mọi $S \subseteq V_1$, $|S| \leq |N_G(S)|$

Chứng minh.

($\Rightarrow$) Giả sử tồn tại một ghép cặp M bao phủ V_1. Do đó, M cũng bao phủ mọi tập con S của V_1. Do đó, với mỗi $v \in S$, tồn tại $w_v \in N_G(v)$ sao cho $vw_v \in M$. Do M là một ghép cặp, với hai đỉnh v, v' phân biệt thuộc $S \subseteq V_1$, ta có $\{v, w_v\} \cap \{v', w_{v'}\} = \emptyset$. Do đó, $\bigcup_{v \in S}\{w_v\} \subseteq \bigcup_{v \in S} N_G(v) = N_G(S)$. Suy ra

$$|N_G(S)| \geq |\bigcup_{v \in S}\{w_v\}| = |\bigcup_{v \in S}\{v\}| = |S|$$

$\square$

Chứng minh (tiếp).

($\Leftarrow$) Ta chứng minh phát biểu $P(m)$ sau đúng với mọi $m \geq 1$ bằng quy nạp mạnh

Cho $|V_1| = m$. Nếu với mọi $S \subseteq V_1$, $|S| \leq |N_G(S)|$ thì tồn tại một ghép cặp $M \subseteq E$ bao phủ V_1

- **Bước cơ sở:** Ta chứng minh $P(1)$ đúng. Thật vậy, do $m = 1$, ta có thể giả sử $V_1 = \{u\}$. Theo giả thiết, $|N_G(u)| \geq |\{u\}| = |V_1| = 1$. Do đó, tồn tại, $v \in N_G(u) \subseteq V_2$, nghĩa là $M = \{uv\}$ là một ghép cặp bao phủ V_1
- **Bước quy nạp:** Giả sử $P(j)$ đúng với mọi $1 \leq j \leq k$, trong đó $k \geq 1$ là số nguyên nào đó. Ta chứng minh $P(k+1)$ đúng. Ta xét hai trường hợp
  1. **Với mọi tập con thực sự $S \neq \emptyset$ của V_1, $|N_G(S)| > |S|$**
  2. **Tồn tại một tập con thực sự $T \neq \emptyset$ của V_1, $|N_G(T)| = |T|$**

□

Chứng minh (tiếp).

(1) **Với mọi tập con thực sự $S \neq \emptyset$ của V_1, $|N_G(S)| > |S|$**

- Lấy một cạnh bất kỳ $e = uv \in E$ với $u \in V_1$ và $v \in V_2$
- Gọi $G' = G - \{u, v\}$. Áp dụng giả thiết quy nạp, tồn tại một ghép cặp M' trong G' bao phủ V_1'. (Tại sao?) Do đó, $M = M' \cup \{uv\}$ là một ghép cặp trong G bao phủ $V_1 = V_1' \cup \{u\}$

(2) **Tồn tại một tập con thực sự $T \neq \emptyset$ của V_1, $|N_G(T)| = |T|$**

- Xét các đồ thị hai phần $H = G[T \cup N_G(T)]$ và $K = G[V_1 - T, V_2 - N_G(T)]$
- Áp dụng giả thiết quy nạp với H và K (Tại sao?), tồn tại một ghép cặp M_1 trong H bao phủ T và một ghép cặp M_2 trong K bao phủ $V_1 - T$. Do đó, $M = M_1 \cup M_2$ là một ghép cặp bao phủ $V_1 = T \cup (V_1 - T)$

$\square$

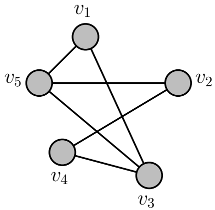
Chú ý:

- Chứng minh trên không cho ta một thuật toán (hiệu quả) để xây dựng một ghép cặp cực đại
- Một chứng minh khác của Định lý Hall (mà chúng ta không thảo luận ở đây) cho ta một thuật toán hiệu quả (trong thời gian đa thức) để tìm một ghép cặp cực đại

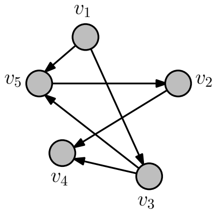
Biểu diễn đồ thị và sự đẳng cấu

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



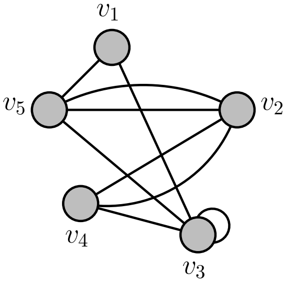
Đỉnh bắt đầu	Đỉnh kết thúc
v_1	v_3, v_5
v_2	v_4
v_3	v_4, v_5
v_4	
v_5	v_2

Biểu diễn đồ thị và sự đẳng cấu

Ma trận kề

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

$$a_{ij} = \begin{cases} m_{ij} & \text{nếu có } m_{ij} \text{ cạnh } v_iv_j \\ 0 & \text{nếu } v_iv_j \notin E \end{cases}$$



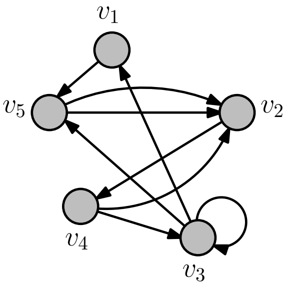
$$\begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 2 & 2 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \end{pmatrix}$$

Biểu diễn đồ thị và sự đẳng cấu

Ma trận kề

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} $(1 \le i, j \le n)$ được định nghĩa như sau

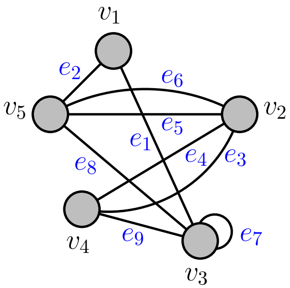
$$a_{ij} = \begin{cases} m_{ij} & \text{nếu có } m_{ij} \text{ cạnh } (v_i, v_j) \\ 0 & \text{nếu } (v_i, v_j) \notin E \end{cases}$$



$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \end{pmatrix}$$

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

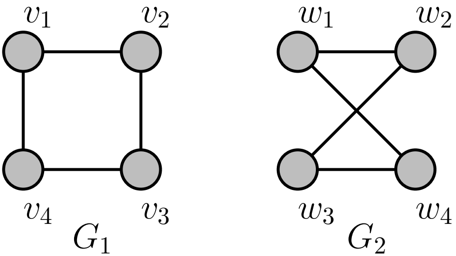
$$a_{ij} = \begin{cases} 1 & \text{nếu cạnh } e_j \text{ liên thuộc với đỉnh } v_i \\ 0 & \text{ngược lại} \end{cases}$$



$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

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 $G_1 \simeq G_2$, nếu tồn tại một song ánh $f : V_1 \to V_2$ 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 $f : V_1 \to V_2$ định nghĩa bởi $f(v_i) = w_i$ $(1 \leq i \leq 4)$ thỏa mãn điều kiện đề ra

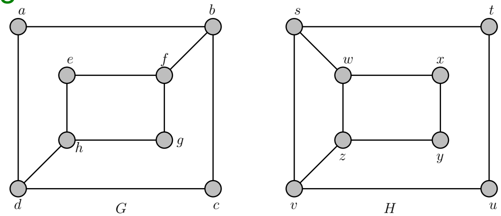
- Một số tính chất hiển nhiên mà các đồ thị đẳng cấu $G_1 = (V_1, E_1)$ và $G_2 = (V_2, E_2)$ cần có
  - $|V_1| = |V_2|$
  - $|E_1| = |E_2|$
  - Với mỗi d, số đỉnh bậc d trong G_1 bằng số đỉnh bậc d trong G_2
  - v.v...
- Thông thường, việc kiểm tra tất cả các song ánh có thể giữa hai tập đỉnh của hai đồ thị G_1, G_2 để xác định xem chúng có đẳng cấu hay không là rất khó khăn: 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

Biểu diễn đồ thị và sự đẳng cấu

Sự đẳng cấu giữa các đồ thị

- 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...)

Ví dụ 13



G và H không đẳng cấu

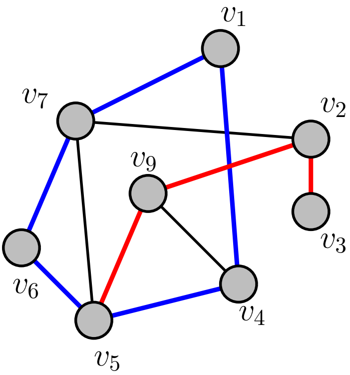
- Do $\deg(a) = 2$, nếu tồn tại một đẳng cấu giữa G và H, a phải tương ứng với một trong bốn đỉnh bậc 2 của H: t, u, x, hoặc y
- Tuy nhiên, mỗi đỉnh trong bốn đỉnh t, u, x, y đều liền kề với một đỉnh bậc hai, trong khi a không thỏa mãn tính chất này trong G

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 $e_1, e_2, \ldots, e_n$ của đồ thị thỏa mãn điều kiện tồn tại một dãy các đỉnh $v_0, v_1, v_2, \ldots, v_{n-1}, v_n$ sao cho $v_0 = u$, $v_n = v$, và e_i có các đầu mút v_{i-1} và v_i, với mọi $i \in \{1, 2, \ldots, n\}$

- Ta nói rằng đường đi bắt đầu với u và kết thúc với v
- 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, 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 $v_0, v_1, \ldots, v_n$

Ví dụ 14



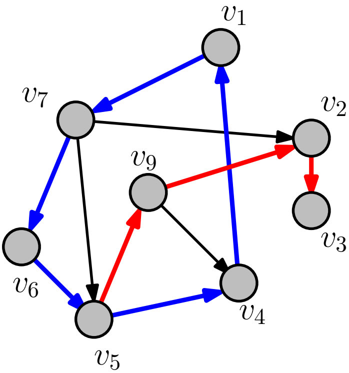
Hình: v_5, v_9, v_2, v_3 là một đường đi độ dài 3 và $v_1, v_4, v_5, v_6, v_7, v_1$ là một chu trình độ dài 5

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
- 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, 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 $v_0, v_1, \dots, v_n$

Ví dụ 15



Hình: v_5, v_9, v_2, v_3 là một đường đi độ dài 3 và $v_1, v_7, v_6, v_5, v_4, v_1$ là một chu trình độ dài 5

- *Độ dài (length)* của một đường đi là số cạnh của đường đi đó
- Một đường đi gọi là *đơn (simple)* nếu nó không chứa cùng một cạnh (cung) nhiều hơn một lần

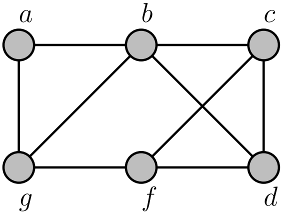
Bài tập 4

Hãy tìm trong đồ thị ở hình bên

(a) *Một đường đi có độ dài* n *với* $n \in \{1, 2, \ldots, 7\}$

(b) *Một đường đi đơn có độ dài* n *với* $n \in \{1, 2, \ldots, 7\}$

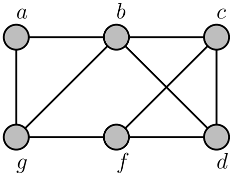
(c) *Một chu trình có độ dài* n *với* $n \in \{3, \ldots, 7\}$



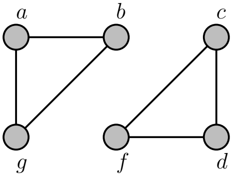
Tính liên thông trong đồ thị

Liên thông trong đồ thị vô hướng

- 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)*



G là đồ thị liên thông



G là đồ thị không liên thông

- *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. Ta gọi các đồ thị con này là các *thành phần liên thông (connected component)* của G
- Cụ thể, 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 của G, H không là đồ thị con thực sự của K
- 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 5

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

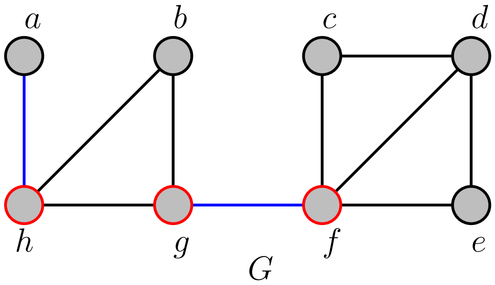
Chứng minh.

- Do G liên thông, luôn tồn tại một đường đi giữa hai đỉnh u, v. Gọi $P = e_1, e_2, \ldots, e_k$ 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, $P' = e_1, e_2, \ldots, e_i, e_{j+1}, \ldots, e_k$ 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

□

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

- Một đồ thị không có đỉnh cắt nào được gọi là *đồ thị không thể tách rời (nonseparable graph)*

Bài tập 5

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 6

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 6 được gọi là một *tập phân tách (separating set (of vertices))* của G

Cho $G = (V, E)$ là một đồ thị vô hướng

- *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
  - $\kappa(K_n) = n - 1$
  - $\kappa(G)$ 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 $0 \leq j \leq k$
  - G là 1-liên thông nếu G là liên thông và có nhiều hơn một đỉnh
  - G là 2-liên thông nếu G không có đỉnh cắt và có ít nhất 3 đỉnh
  - Nếu *xóa đi tối đa $k-1$ đỉnh bất kỳ* từ G thì đồ thị thu được luôn là đồ thị liên thông

Bài tập 7

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 7 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*
- G được gọi là *k-liên thông cạnh (k-edge connected)* nếu $\lambda(G) \geq k$.
    - $\lambda(G) = 0$ nếu G không liên thông hoặc chỉ có một đỉnh
    - $\lambda(K_n) = n - 1$
    - 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

Bài tập 8

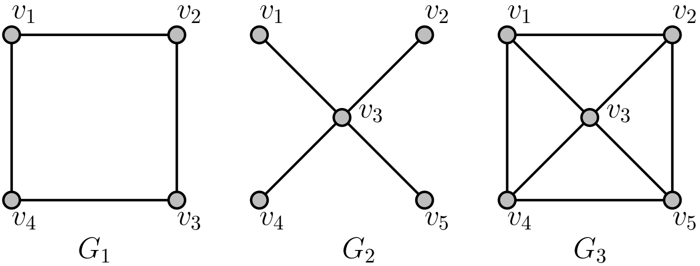
Chứng minh rằng với mọi đồ thị vô hướng liên thông $G = (V, E)$

$$\kappa(G) \leq \min_{v \in V} \deg_G(v) \quad (1)$$

$$\lambda(G) \leq \min_{v \in V} \deg_G(v) \quad (2)$$

Bài tập 9

Xác định $\kappa(G_i)$ *và* $\lambda(G_i)$ *trong các đồ thị* G_i *với* $i = 1, 2, 3$ *sau*



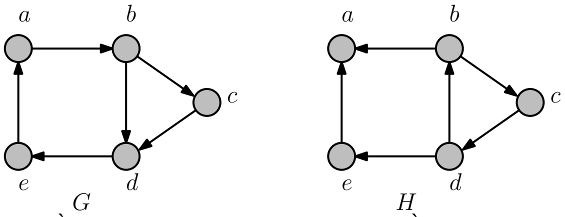
Tính liên thông trong đồ thị

Liên thông trong đồ thị có hướng

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 16



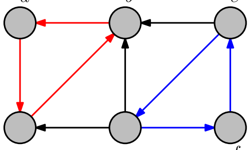
Hình: G là đồ thị liên thông mạnh. H không là đồ thị liên thông mạnh nhưng là đồ thị liên thông yếu

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 17

- 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

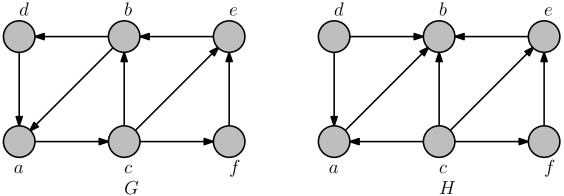


G

Tính liên thông trong đồ thị

Liên thông trong đồ thị có hướng

- 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.

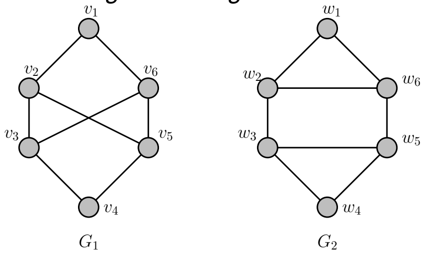


Hình: G là một đồ thị có hướng và có chu trình. H là một đồ thị có hướng và không có chu trình

- **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 10

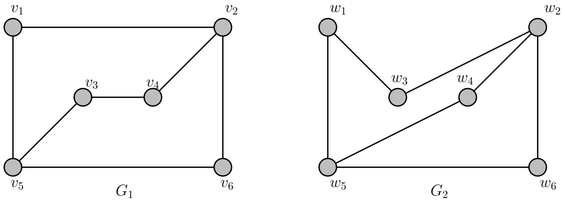
Các đồ thị sau có đẳng cấu không? Vì sao?



- Chúng ta cũng có thể sử dụng đường đi để tìm các ánh xạ giữa hai đồ thị đẳng cấu

Bài tập 11

Các đồ thị sau có đẳng cấu không? Vì sao?



Định lý 6

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 6 đúng với $r = 1$
- **Bước quy nạp:** Giả sử Định lý 6 đúng với mọi $1 \leq r \leq k$. Ta chứng minh Định lý 6 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 $\{v_\ell, v_j\}$.

60

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

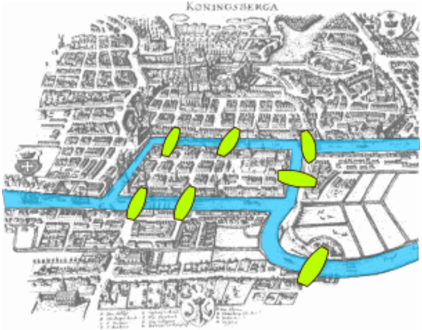
Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn

Nội dung

Đường đi Euler và Đường đi Hamilton

Bảy cây cầu ở Königsberg

Hình: Leonhard Euler 1707–1783 (Wikipedia)



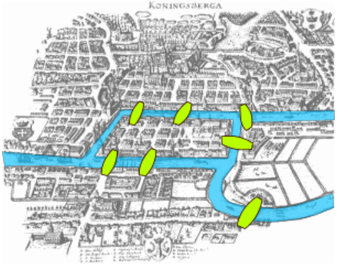
Hình: Bản đồ Königsberg cũ (Wikipedia)

Bảy cây cầu ở Königsberg

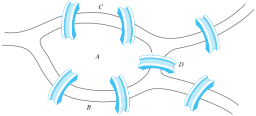
Tìm một tuyến đường đi qua mỗi cây cầu chính xác một lần và quay lại vị trí xuất phát

Đường đi Euler và Đường đi Hamilton

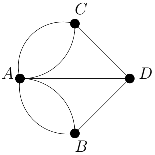
Bảy cây cầu ở Königsberg



(a) Bản đồ Königsberg cũ (Wikipedia)



(b) Bản đồ Königsberg cũ đơn giản hóa



(c) Đồ thị tương ứng

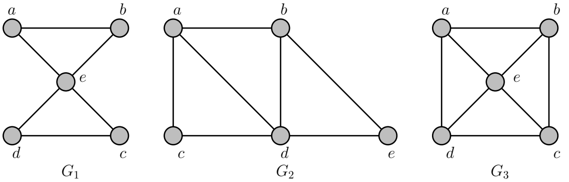
- Đồ thị tương ứng:
  - Mỗi vùng đất ứng với một đỉnh
  - Mỗi cây cầu nối hai vùng đất ứng với một cạnh
- Tìm chu trình đơn trong đồ thị chứa tất cả các cạnh

Đường đi Euler và Đường đi Hamilton

Đường đi Euler

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

Định lý 1

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 $x_0 = x_m = u$.

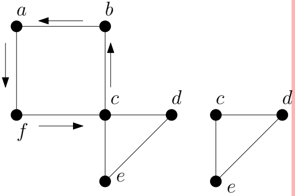
- Với $v = x_i$ $(2 \leq i \leq m - 1)$: chu trình đi vào v qua e_i và đi ra qua e_{i+1}
- Với $u = x_0 = x_m$: chu trình đi ra u qua e_1 và trở lại u qua e_m

□

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)**)

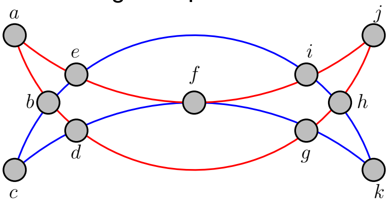
- Xuất phát từ đỉnh $x_0 = a$ bất kỳ
- Xây dựng một đường đi đơn bằng cách chọn tùy ý các cạnh $x_0x_1, x_1x_2, \dots, x_{k-1}x_k$ để thêm vào đường đi cho đến khi không chọn được nữa
- Do bậc của mỗi đỉnh là chẵn, với mỗi đỉnh x_i, ta luôn có thể đi vào từ cạnh $x_{i-1}x_i$ và đi ra từ cạnh x_ix_{i+1}. 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 $O(|E|)$)

Ví dụ 2

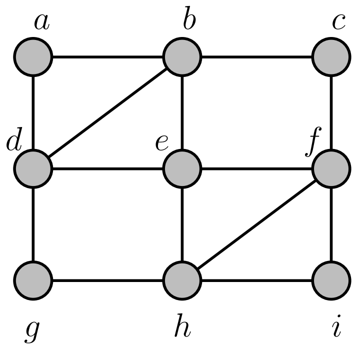
Tìm chu trình Euler trong đồ thị sau



- Bắt đầu từ $x_0 = a$, chọn tùy ý các cạnh x_0x_1, x_1x_2, $\dots$, $x_{k-1}x_k$, x_ka. 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_lc. 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:
  - $ae, ei, ih, hk, kg, gf, fd, dc, cb, be, ef, fi, ij, jh, hg, gd, db, ba$
  - $ae, ef, fd, dc, cb, be, ei, ih, hk, kg, gf, fi, ij, jh, hg, gd, db, ba$
  - $\dots$

Bài tập 2

Tìm chu trình Euler trong đồ thị sau.



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

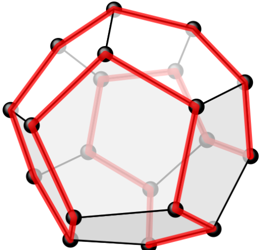
($\Leftarrow$) Giả sử G có chính xác hai đỉnh bậc lẻ u, v
- Tìm chu trình Euler của đồ thị $G + uv$
- Xóa cạnh uv trong chu trình để thu được đường đi Euler trong G

$\square$

Đường đi Euler và Đường đi Hamilton

Trò chơi “Vòng quanh thế giới” (1857)

Hình: Sir William Rowan Hamilton 1805–1865 (Wikipedia)



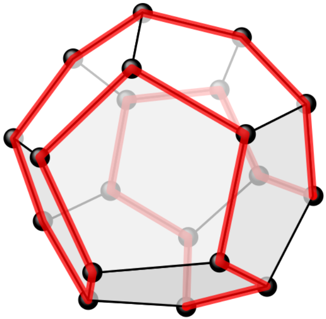
Hình: Trò chơi “Vòng quanh thế giới” (Wikipedia)

Trò chơi “Vòng quanh thế giới”

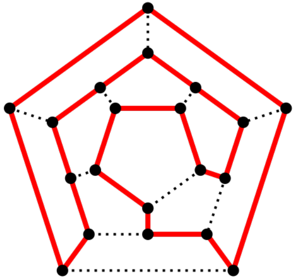
Mỗi đỉnh trong 20 đỉnh của khối 12 mặt đại diện cho một thành phố. Tìm đường đi xuất phát từ một đỉnh dọc theo các cạnh của khối, ghé thăm mỗi đỉnh còn lại một lần, và quay lại vị trí ban đầu

Đường đi Euler và Đường đi Hamilton

Trò chơi “Vòng quanh thế giới” (1857)



(a) Trò chơi “Vòng quanh thế giới” (Wikipedia)



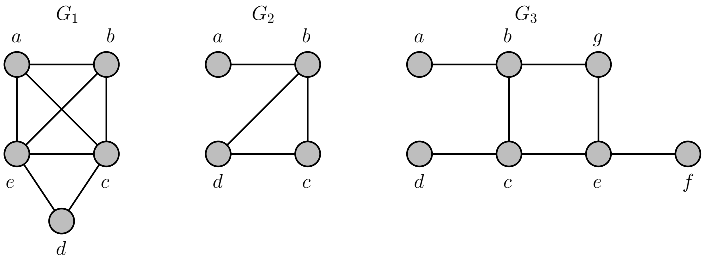
(b) Đồ thị đẳng cấu với khối 12 mặt (Wikipedia)

Cho $G = (V, E)$ là một đồ thị vô hướng

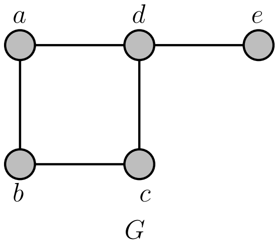
- Một *đường đi Hamilton* trong G là một đường đi đơn $x_0, x_1, \ldots, x_{n-1}, x_n$ thỏa mãn điều kiện $V = \{x_0, x_1, \ldots, x_n\}$ và $x_i \neq x_j$ với $0 \leq i < j \leq n$
- Một *chu trình Hamilton* trong G là một chu trình đơn $x_0, x_1, x_{n-1}, x_n, x_0$ thỏa mãn điều kiện x_0, x_1, x_{n-1}, x_n là một đường đi Hamilton

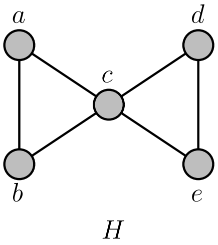
Bài tập 3

Các đồ thị sau có chu trình/đường đi Hamilton không?



- *Chưa có* điều kiện cần và đủ 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*



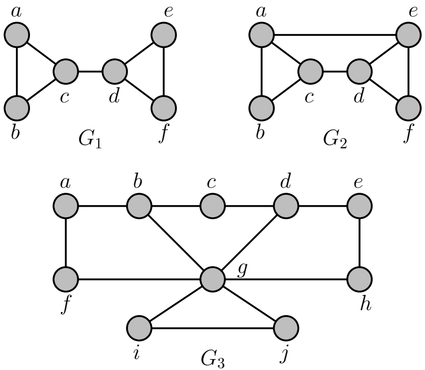


Bài tập 4

Hãy cho ví dụ về một đồ thị mà chu trình Euler của nó cũng là chu trình Hamilton

Bài tập 5

Trong các đồ thị sau, đồ thị nào có chu trình Hamilton? Tại sao?



Định lý 3: 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 6 (($\star$) Chứng minh Định lý Dirac)

(a) *Dễ 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_iv_{i+1} của P thỏa mãn $v_iv_k \in E$. Tương tự, do $\deg(v_0) \geq n/2$, có ít nhất $n/2$ cạnh phân biệt v_jv_{j+1} của P thỏa mãn $v_0v_{j+1} \in E$*
- *Do P có ít hơn n cạnh, tồn tại một cạnh v_qv_{q+1} thỏa mãn đồng thời hai điều kiện trên: $v_qv_k \in E$ và $v_0v_{q+1} \in E$*

(d) *P chứa tất cả các đỉnh của G (Tại sao?)*

Định lý 4: 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 7

Chứng minh Định lý Dirac (Định lý 3) bằng cách sử dụng Định lý Ore

Bài tập 8

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

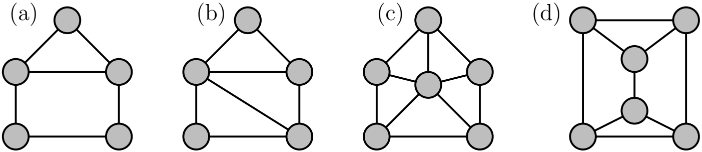
Bài tập 9

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?*

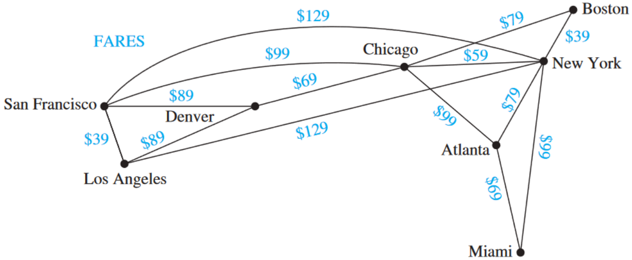
(iii) *đồ thị có chu trình Hamilton không?*



Bài toán đường đi ngắn nhất

Đồ thị có trọng số

- Một *đồ thị có trọng số (weighted graph)* $G = (V, E, w)$ gồm tập đỉnh V, tập cạnh E, và một hàm $w : E \to \mathbb{R}$ gán mỗi cạnh (cung) $e \in E$ bởi một số thực $w(e)$ gọi là *trọng số (weight)* của cạnh (cung) e
- Trong bài giảng, chúng ta chỉ xét các đồ thị có *trọng số dương*, nghĩa là, $w : E \to \mathbb{R}^+$



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)

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 *chiều 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 $V = \{v_0 = a, v_1, \dots, v_n = z\}$, $w : [V]^2 \to \mathbb{R}^+ \cup \{\infty\}$ với $w(v_i, v_j) = \infty$ nếu $v_i v_j \notin E$
- **Output:** Khoảng cách $d_G(a, z)$

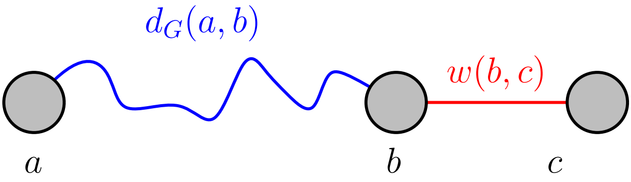
Bài toán đường đi ngắn nhất

Thuật toán Dijkstra

Bài toán đường đi ngắn nhất

- **Input:** Đơn đồ thị vô hướng $G = (V, E, w)$ trong đó $V = \{v_0 = a, v_1, \dots, v_n = z\}$, $w : [V]^2 \to \mathbb{R}^+ \cup \{\infty\}$ với $w(v_i, v_j) = \infty$ nếu $v_iv_j \notin E$
- **Output:** Khoảng cách $d_G(a, z)$

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 kề với c* bằng *khoảng cách giữa a và b* cộng với *trọng số cạnh nối b và c*

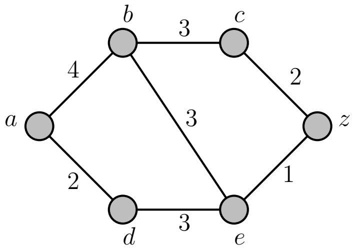


Thuật toán Dijkstra

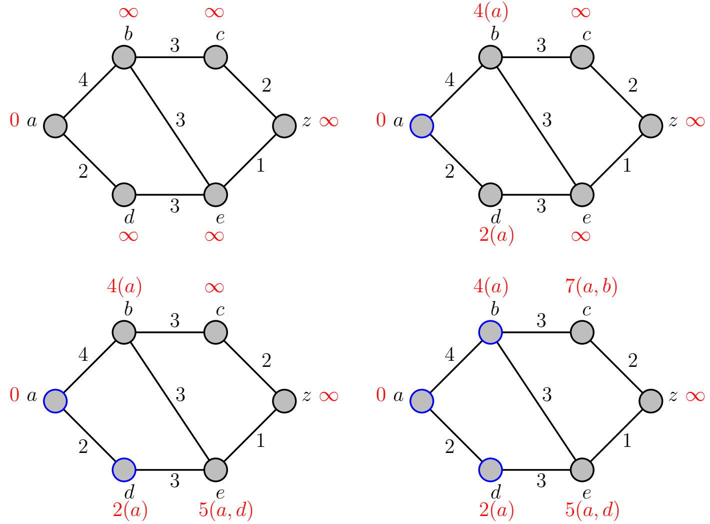
- Khởi tạo: Gán nhãn $L(a) := 0$, $L(v_i) := \infty$ với mọi $v_i \neq a$, và lấy một tập $S := \emptyset$.
- Trong khi $z \notin S$, lặp lại các bước sau:
  - Gọi u là đỉnh không thuộc S với $L(u)$ nhỏ nhất. Thêm u vào S.
  - Với mọi đỉnh v không thuộc S
    - Nếu $L(u) + w(u, v) < L(v)$ thì gán $L(v) := L(u) + w(u, v)$ (Sửa đổi nhãn của các đỉnh không thuộc S)
- Cuối cùng, $L(z)$ là độ dài đường đi ngắn nhất (khoảng cách) từ a đến z.

Ví dụ 3

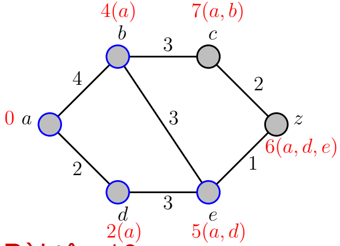
Sử dụng thuật toán Dijkstra để tìm khoảng cách giữa hai đỉnh a và z trong đồ thị sau

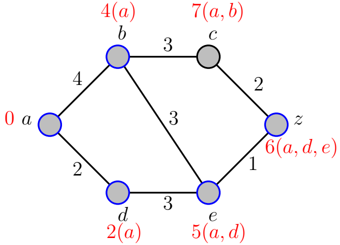


Bài toán đường đi ngắn nhất
Thuật toán Dijkstra



Bài toán đường đi ngắn nhất
Thuật toán Dijkstra

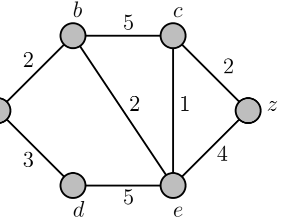




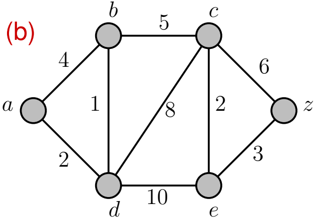
Bài tập 10

Áp dụng thuật toán Dijkstra để tìm khoảng cách từ a đến z trong mỗi đồ thị sau

(a)



(b)



Bài toán đường đi ngắn nhất

Thuật toán Dijkstra

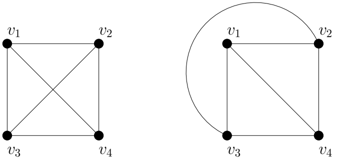
- Tính đúng đắn của thuật toán Dijkstra có thể được chứng minh bằng cách *sử dụng bất biến vòng lặp*. Cụ thể, trước mỗi lần lặp, hàm L và tập S thỏa mãn các tính chất sau:
  (1) Với mọi $v \in S$, $L(v)$ là khoảng cách từ a đến v
  (2) Với mọi $v \in V - S$, $L(v)$ là độ dài đường đi ngắn nhất từ a đến v **chỉ qua các đỉnh thuộc** $S \cup \{v\}$
  (3) Với mọi $u \in S$ và $v \in V - S$, $L(u) \leq L(v)$
- Thông thường, thuật toán Dijkstra chạy trong thời gian $O(n^2)$, 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 của cách lập trình này được thể hiện khi chạy với các "đồ thị thưa" (sparse graph) cực lớn (các đồ thị có m rất nhỏ so với n^2)
- 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

Đồ thị phẳng
Định nghĩa và khái niệm

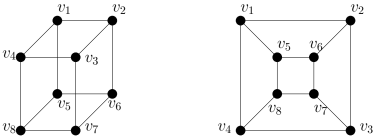
- Một đồ thị vô hướng được gọi là *đồ thị phẳng (planar graph)* nếu nó có thể được vẽ trên mặt phẳng sao cho không có hai cạnh nào cắt nhau (ở một điểm không phải là đầu mút của cạnh).
- Hình vẽ như thế được gọi là một *biểu diễn phẳng (planar representation)* của đồ thị.

Ví dụ 4

K_4

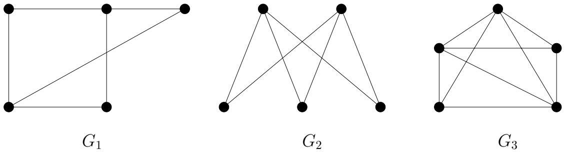


Q_3



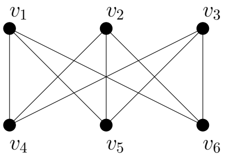
Bài tập 11

Tìm một biểu diễn phẳng của các đồ thị phẳng sau



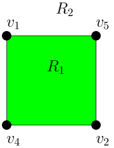
Ví dụ 5

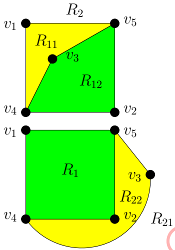
$K_{3,3}$ không là đồ thị phẳng



Ta chứng minh khẳng định trên 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?





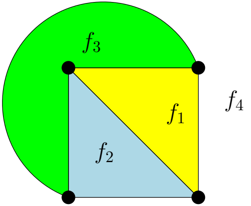
Đồ thị phẳng
Công thức Euler

- 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 6

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)

Định lý 5: 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}

Hệ quả 6

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 $\sum_{\text{miền } f \text{ của } G} \deg(f) \leq 2m$. Suy ra $3r \leq 2m$.
- Áp dụng công thức Euler, ta có $2 = n - m + r \leq n - m + 2m/3$, suy ra $m \leq 3n - 6$.

□

Bài tập 12

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 13

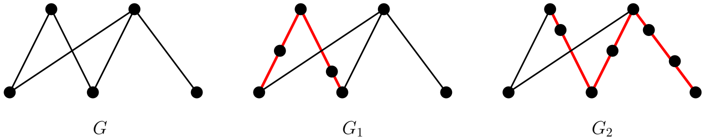
Chứng minh K_5 không là đồ thị phẳng

Bài tập 14

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

- 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 7

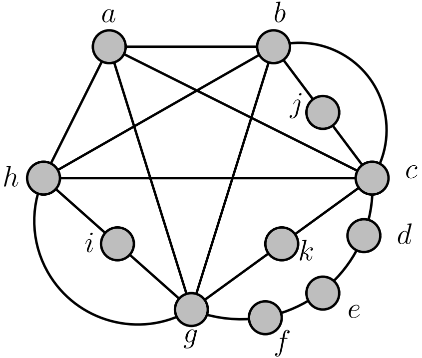


Định lý 7: Định lý Kuratowski

G là đồ thị phẳng khi và chỉ khi nó không chứa bất kỳ đồ thị nào đồng phôi với K_5 hoặc $K_{3,3}$

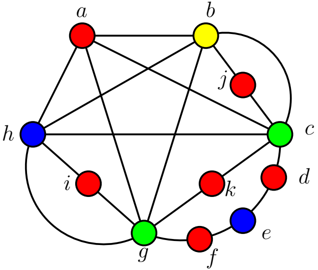
Bài tập 15

Sử dụng Định lý Kuratowski, hãy chứng minh đồ thị sau không là đồ thị phẳng



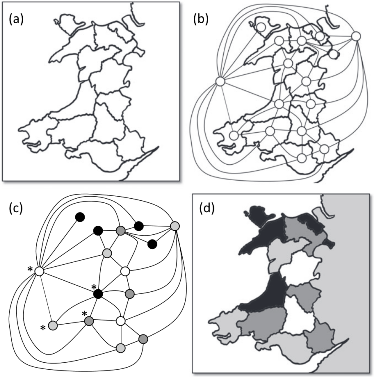
Cho đơn đồ thị vô hướng $G = (V, E)$

- *Tô màu* một đồ thị đơn là sự gán màu cho các đỉnh của đồ thị sao cho không có hai đỉnh liền 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 $f : V \to \{1, 2, \dots, k\}$ 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

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



Hình: [Lewis 2021], Hình 1.7

- Phỏng đoán của Guthrie được cho là phát biểu đầu tiên của *Định lý bốn màu (Four Color Theorem)*

Định lý 8: 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

- 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.
  - Chỉ ra rằng không thể dùng ít hơn k màu để tô màu các đỉnh của G.
- 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 $\chi(G) = 2$ (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 16

Tính $\chi(W_n)$, $\chi(K_{m,n})$, và $\chi(Q_n)$

Gọi $\Delta(G)$ là *bậc lớn nhất của các đỉnh của đồ thị G*

Định lý 9

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 $\{1, \ldots, \Delta(G) + 1\}$ là như sau:

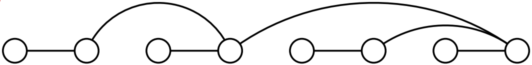
1. Gán nhãn $v_1, v_2, \ldots, v_n$ cho các đỉnh của G một cách tùy ý
2. Với i từ 1 đến n, tô màu đỉnh v_i bằng màu nhỏ nhất trong số các màu chưa được tô cho bất kỳ đỉnh nào trong $N(v_i)$

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 □

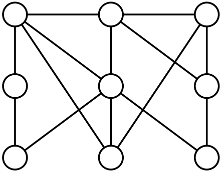
Bài tập 17

Sử dụng thuật toán ở Định lý 9 để tô màu các đồ thị sau

(a)



(b)



Trên thực tế, với phần lớn các đồ thị, chỉ cần $\Delta(G)$ màu là đủ

Định lý 10: Định lý Brook

Nếu G không phải là một chu trình độ dài lẻ hoặc một đồ thị đầy đủ thì $\chi(G) \leq \Delta(G)$

Bổ đề 11

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 14, 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

□

Định lý 12

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 $v_1, v_2, \ldots, v_n$ của G thỏa mãn Bổ đề 11: mỗi đỉnh có tối đa 5 đỉnh kề đứng trước nó
2. Áp dụng thuật toán tham lam ở Định lý 9 với thứ tự đỉnh tìm được ở Bước 1

Chú ý rằng

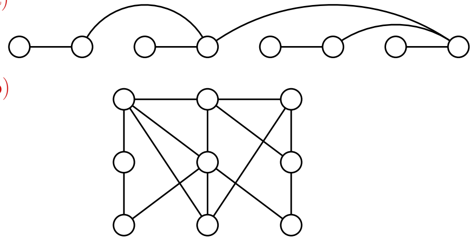
- Khuyên và cạnh song song (nếu có) không ảnh hưởng gì đến quá trình tô màu
- Nếu đồ thị phẳng đã cho không liên thông, ta có thể áp dụng quá trình tô màu riêng biệt cho từng thành phần liên thông

Bài tập 18

Sử dụng thuật toán ở Định lý 12 để tô màu các đồ thị phẳng sau bằng 6 màu $\{1, 2, 3, 4, 5, 6\}$

(a)

(b)



Nếu có thể, hãy tìm một cách tô màu các đồ thị trên bằng 5 màu hoặc ít hơn

Bài tập 19

*Chứng minh Định lý 12 bằng phương pháp quy nạp. (**Gợi ý:** Xem lại chứng minh Bổ đề 11)*

Tài liệu tham khảo

- 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.
- 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.

Tài liệu tham khảo (tiếp)

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.

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

Giới thiệu

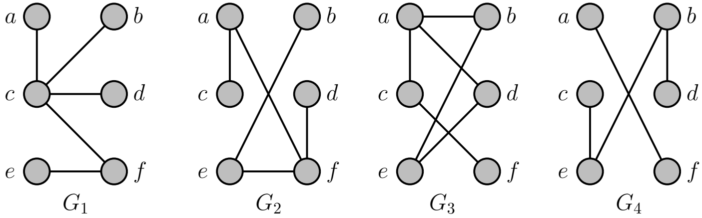
Một số tính chất của cây

Cây có gốc

Cây bao trùm

Cây bao trùm nhỏ nhất

- 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

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

$\square$

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 □

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

□

Bài tập 1

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 2

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

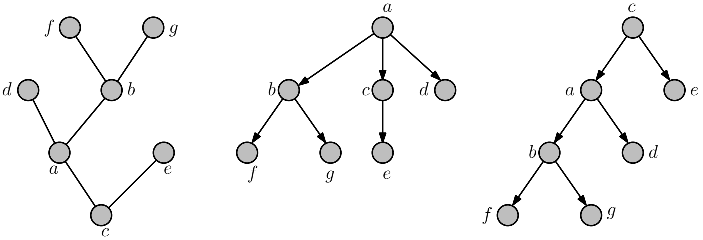
Bài tập 3

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

Bài tập 4

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

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

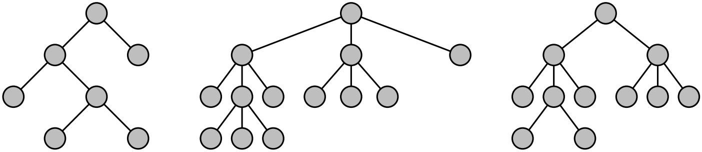
Thông thường, một cây có gốc (T, r) được vẽ với *đỉnh gốc ở trên đỉnh của đồ thị*. Khi đó, ta có thể *bỏ qua các hướng của các cạnh*, do việc lựa chọn đỉnh gốc đã xác định các hướng này

- 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

- Một cây có gốc T được gọi là một *cây m-phân (m-ary tree)* 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 (full m-ary tree)*
- Cây m-phân với $m = 2$ được gọi là *cây nhị phân (binary tree)*

Bài tập 5

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?



T_1 T_2 T_3

Cây

Cây có gốc

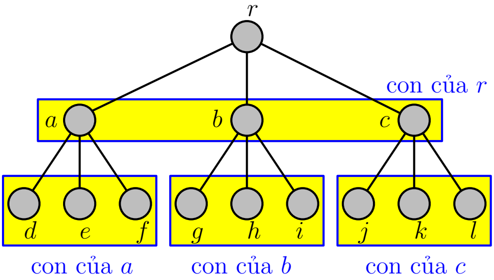
- Một *cây có gốc được sắp thứ tự (ordered rooted tree)* là một cây có gốc trong đó các đỉnh con của mỗi đỉnh được sắp xếp theo một thứ tự nào đó
- Ta thường vẽ cây có gốc được sắp thứ tự theo cách thông thường với hiểu ngầm rằng *các đỉnh con của mỗi đỉnh được sắp xếp theo thứ tự từ trái sang phải*
- 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 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

Với mọi $m \geq 1$, mọi cây m-phân đầy đủ 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



- *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
- 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

Định lý 5

Với mọi $m \geq 1$, một cây m-phân đầy đủ với

(i) *n đỉnh có $i = (n-1)/m$ đỉnh trong và $l = ((m-1)n+1)/m$ đỉnh lá;*

(ii) *i đỉnh trong có $n = mi + 1$ đỉnh và $l = (m-1)i + 1$ đỉnh lá;*

(iii) *l đỉnh lá có $n = (ml-1)/(m-1)$ đỉnh và $i = (l-1)/(m-1)$ đỉnh trong*

Bài tập 6

Sử dụng Định lý 4, hãy chứng minh Định lý 5

Bài tập 7

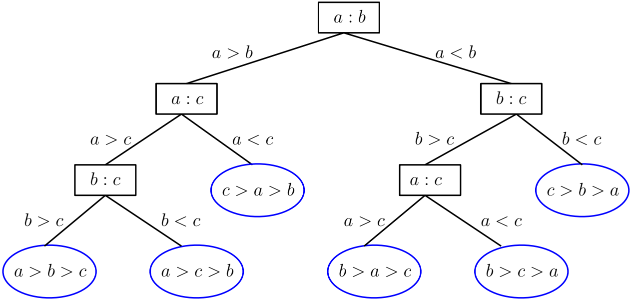
*Chứng minh rằng với mọi $m \geq 1$, có nhiều nhất m^h đỉnh lá trong một cây m-phân có độ cao h. (**Gợi ý:** Quy nạp theo h)*

Một ứng dụng 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 $\lceil \log_2(n!) \rceil$ phép so sánh trong trường hợp xấu nhất

- Do $\log_2(n!) = \Theta(n \log n)$, *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 thời gian 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 dài đường đi từ gốc đến nút lá tương ứng*

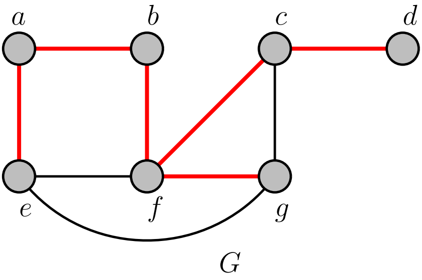


Hình: Mô hình thuật toán sắp xếp dãy ba phần tử a, b, c bằng cây quyết định

Bài tập 8

Với mọi $m \geq 1$, chứng minh rằng trong một cây m-phân có l đỉnh lá và có độ cao h, ta có $h \geq \lceil \log_m l \rceil$

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 cho đến khi không có chu trình trong G

Do G có hữu hạn số đỉnh, cuối cùng ta thu được một đồ thị liên thông không có chu trình chứa tất cả các đỉnh của G—một cây bao trùm của G □

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

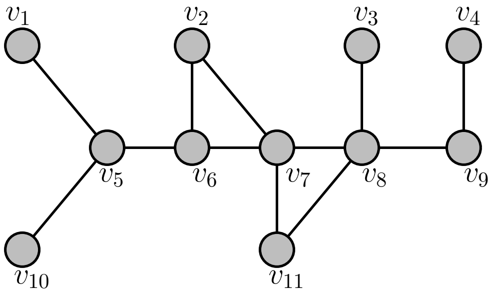
procedure `dfs`(G)**:**
  $T :=$ cây có chính xác một đỉnh v_1
  `visit`(v_1)

procedure `visit`(v*: một đỉnh của* G)**:**
  for *mỗi đỉnh* w *liền kề với* v *và chưa thuộc* T **do**
    Thêm w và cạnh vw vào T
    `visit`(w)

return T

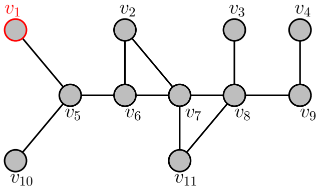
Ví dụ 1

Tìm một cây bao trùm T của đồ thị $G = (V, E)$ sau bằng thuật toán duyệt theo chiều sâu

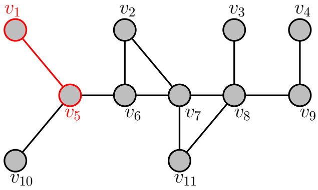


G

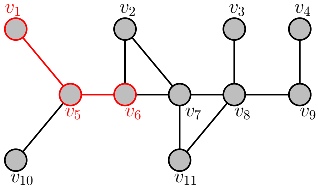
Cây
Cây bao trùm



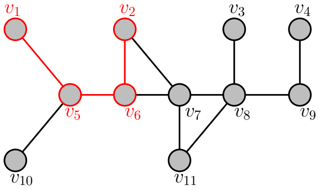
G



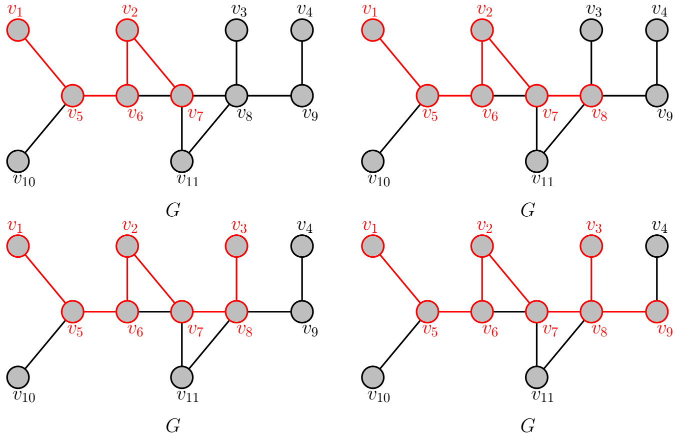
G

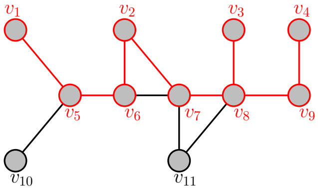


G

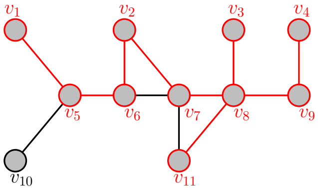


G

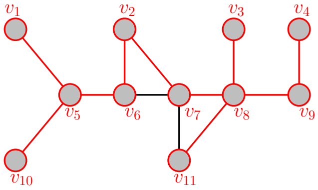

v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
v_{10}
v_{11}
G



G



G



G

Thứ tự các đỉnh được thêm vào T (hay thứ tự các đỉnh thực hiện thủ tục `visit()`) lần lượt là: v_1, v_5, v_6, v_2, v_7, v_8, v_3, v_9, v_4, v_{11}, v_{10}

- 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

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

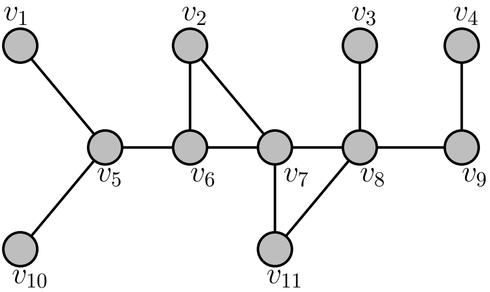
Thuật toán 2: Duyệt theo chiều rộng

Input: G: đơn đồ thị vô hướng liên thông với các đỉnh $v_1, v_2, \dots, v_n$
Output: Cây bao trùm T

```
T := cây chỉ chứa duy nhất một đỉnh v1
L := danh sách rỗng
Thêm v1 vào danh sách L các đỉnh chưa xét
while L khác rỗng do
    Bỏ đi đỉnh thứ nhất v từ L
    for mỗi đỉnh w liền kề với v do
        if w không thuộc L và w không thuộc T then
            Thêm w vào cuối danh sách L
            Thêm w và cạnh vw vào T
return T
```

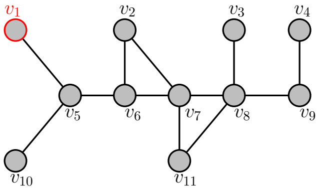
Ví dụ 2

Tìm một cây bao trùm T của đồ thị $G = (V, E)$ sau bằng thuật toán duyệt theo chiều rộng



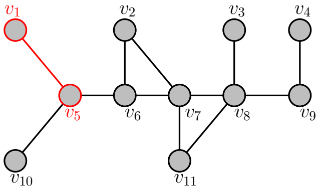
G

$L = (v_1)$



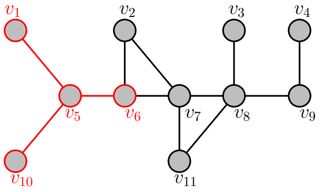
G

$L = (v_5)$



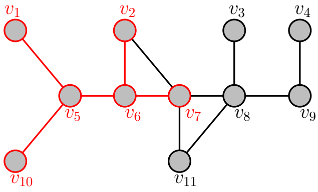
G

$L = (v_6, v_{10})$



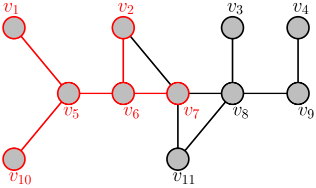
G

$L = (v_{10}, v_2, v_7)$



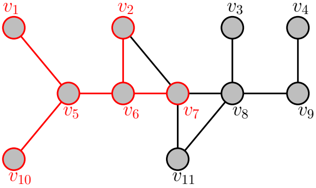
G

$L = (v_2, v_7)$



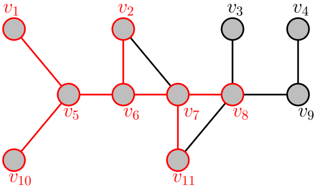
G

$L = (v_7)$



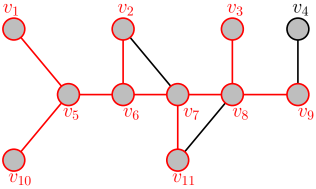
G

$L = (v_8, v_{11})$



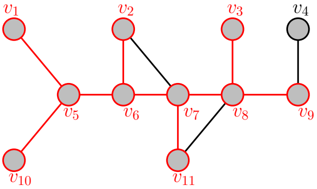
G

$L = (v_{11}, v_3, v_9)$



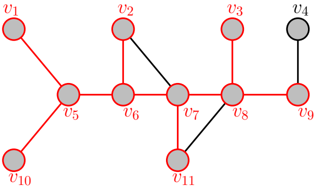
G

$L = (v_3, v_9)$



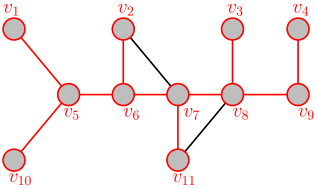
G

$L = (v_9)$



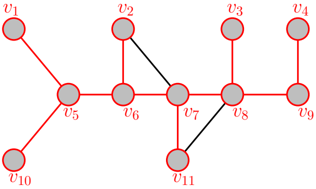
G

$L = (v_4)$



G

$L = ()$



G

- 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 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...

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

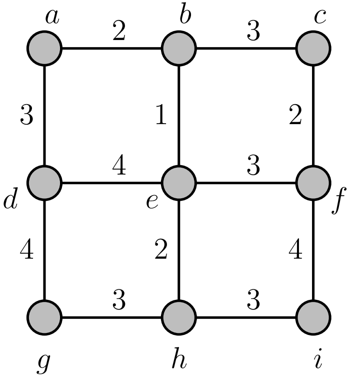
Thuật toán 3: Thuật toán Prim

Input: $G = (V, E, w)$: đồ thị liên thông vô hướng có trọng số
Output: Một cây bao trùm nhỏ nhất T của G

$T :=$ một cạnh có trọng số nhỏ nhất
for $i := 1$ **to** $n - 2$ **do**
  $e :=$ một cạnh có trọng số nhỏ nhất liên thuộc với một đỉnh của T thỏa mãn $T + e$ không có chu trình
  $T := T + e$
return T

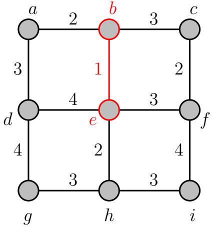
Ví dụ 3

Tìm một cây bao trùm nhỏ nhất T của đồ thị $G = (V, E, w)$ sau bằng thuật toán Prim

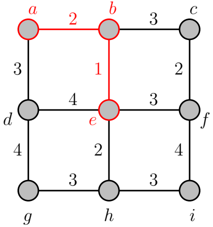


G

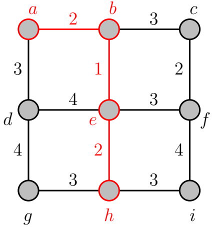
Cây
Cây bao trùm nhỏ nhất



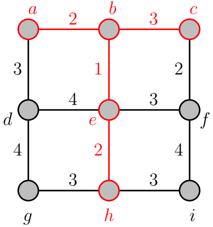
G



G

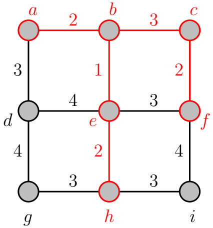


G

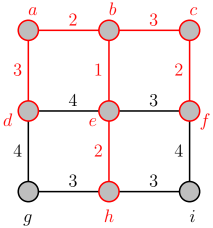


G

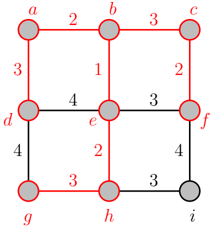
Cây
Cây bao trùm nhỏ nhất



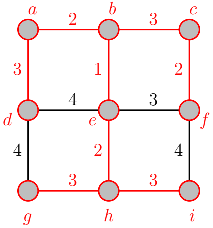
G



G



G



G

Định lý 8

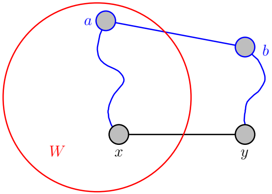
Nếu T là một cây bao trùm xuất ra bởi thuật toán Prim với đồ thị đầu vào $G = (V, E, w)$ thì T là một cây bao trùm nhỏ nhất của G

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 liền kề 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

$T :=$ đồ thị rỗng (không có đỉnh và cạnh)

for $i := 1$ **to** $n - 1$ **do**

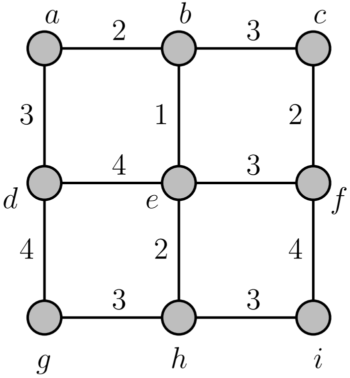
  $e :=$ một cạnh có trọng số nhỏ nhất thỏa mãn $T + e$ không có chu trình

  $T := T + e$

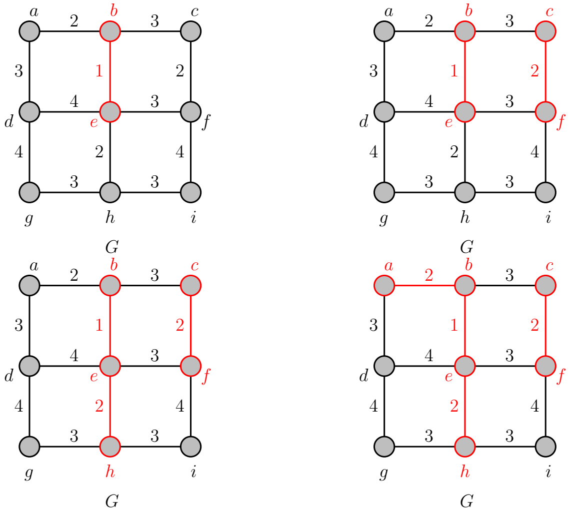
return T

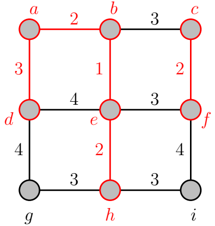
Ví dụ 4

Tìm một cây bao trùm nhỏ nhất T của đồ thị $G = (V, E, w)$ sau bằng thuật toán Kruskal



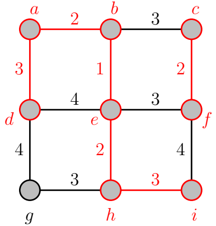
G


a
2
b
3
c
3
1
2
4
3
d
e
f
4
2
4
3
3
g
h
i
G



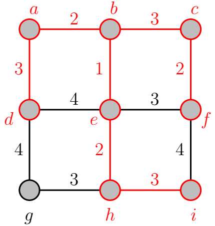
a
2
b
3
c
3
1
2
d
4
e
3
f
4
2
4
3
3
g
h
i

G



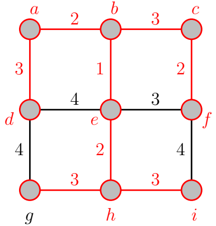
a
2
b
3
c
3
1
2
d
4
e
3
f
4
2
4
3
3
g
h
i

G



a
2
b
3
c
3
1
2
d
4
e
3
f
4
2
4
3
3
g
h
i

G



a
2
b
3
c
3
1
2
d
4
e
3
f
4
2
4
3
3
g
h
i

G

Bài tập 9

*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 $m = |E|$

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

Nội dung

Hàm Boole

Biểu diễn các hàm Boole

Các cổng lôgic

Rút gọn hàm Boole

Hàm Boole

- *Đại số Boole (Boolean Algebra)* đưa ra các phép toán và quy tắc làm việc với tập $\{0, 1\}$ (ứng với $\{\mathbf{F}, \mathbf{T}\}$)
- 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 lôgic $(\mathbf{T} \wedge \mathbf{T}) \vee \neg\mathbf{F} \equiv \mathbf{T}$ *sang một đẳng thức trong Đại số Boole.*

Hàm Boole

Giả sử $B = \{0, 1\}$

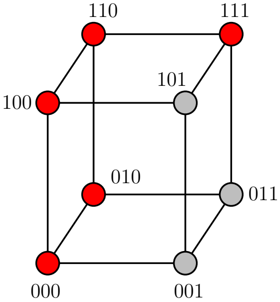
- $B^n = \{(x_1, x_2, \dots, x_n) \mid x_i \in B$ với $1 \leq i \leq n\}$
- Một biến x được gọi là một *biến Boole (Boolean variable)* nếu nó chỉ nhận các giá trị từ B
- Một hàm $F : B^n \to B$ được gọi là một *hàm Boole bậc n (Boolean function of degree n)*. Các giá trị của một hàm Boole thường được 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_1 và E_2 là các biểu thức Boole thì $\overline{E}$, $(E_1 + E_2)$, 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 $F(x_1, \dots, x_n) = 1$

Hàm Boole

Ví dụ 1

Hàm $F(x, y, z) = xy + \overline{z}$

x	y	z	xy	$\overline{z}$	$F(x, y, z)$
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



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 3

Có tất cả bao nhiêu hàm Boole bậc n khác nhau?

Hàm Boole

Một số hằng đẳng thức quan trọng trong Đại số Boole

Tên gọi	Đẳng thức
Luật phần bù kép (Law of the double complement)	$\overline{\overline{x}} = x$
Luật lũy đẳng (Idempotent laws)	$x + x = x$ $x \cdot x = x$
Luật đồng nhất (Identity laws)	$x + 0 = x$ $x \cdot 1 = x$
Luật nuốt (Domination laws)	$x + 1 = 1$ $x \cdot 0 = 0$
Luật giao hoán (Commutative laws)	$x + y = y + x$ $xy = yx$
Luật kết hợp (Associative laws)	$(x + y) + z = x + (y + z)$ $x(yz) = (xy)z$
Luật phân phối (Distributive laws)	$x + yz = (x + y)(x + z)$ $x(y + z) = xy + xz$

Hàm Boole

Một số hằng đẳng thức quan trọng trong Đại số Boole (tiếp)

Tên gọi	Đẳng thức
Luật De Morgan (De Morgan laws)	$\overline{(xy)} = \overline{x} + \overline{y}$ $\overline{(x+y)} = \overline{x}\,\overline{y}$
Luật hấp thụ (Absorption laws)	$x + xy = x$ $x(x+y) = x$
Tính chất đơn vị (unit property)	$x + \overline{x} = 1$
Tính chất không (zero property)	$x\overline{x} = 0$

Bài tập 4

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 5

Bằng cách sử dụng các đẳng thức quan trọng khác trong bảng, hãy chứng minh luật hấp thụ $x(x+y) = x$

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)
  - $+ \Rightarrow \cdot$
  - $\cdot \Rightarrow +$
  - $0 \Rightarrow 1$
  - $1 \Rightarrow 0$
- 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 $\overline{x} \cdot 1 + (\overline{y} + z)$ là $(\overline{x} + 0) \cdot (\overline{y} \cdot z)$

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

- 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) $^{-}$, ký hiệu $[B, \vee, \wedge, ^{-}, 0, 1]$, 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): $x \vee \overline{x} = 1$ và $x \wedge \overline{x} = 0$
- 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 $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$

Hàm Boole

- $[B, +, \cdot, \bar{}, 0, 1]$, trong đó $B = \{0, 1\}$ và $+, \cdot, \bar{}$ 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 lôgic và $\vee, \wedge, \neg$ là các toán tử lôgic, là một đại số Boole
- $[\mathcal{P}(U), \cup, \cap, \overline{}, \emptyset, U]$, trong đó U là tập vũ trụ nào đó, $\mathcal{P}(U)$ là tập tất cả các tập con của U, và $\cup, \cap, \overline{}$ là các toán tử tập hợp, là một đại số Boole

Hàm Boole

Ví dụ 4

Sử dụng định nghĩa, chứng minh rằng trong một đại số Boole $[B, \vee, \wedge, ^{-}, 0, 1]$, với mọi $x \in B$ ta có $x \vee x = x$ và $x \wedge x = x$ (luật lũy đẳng (idempotent laws))

$$
\begin{aligned}
x \vee x &= (x \vee x) \wedge 1 && \text{luật đồng nhất} \\
&= (x \vee x) \wedge (x \vee \overline{x}) && \text{luật bù} \\
&= x \vee (x \wedge \overline{x}) && \text{luật phân phối} \\
&= x \vee 0 && \text{luật bù} \\
&= x && \text{luật đồng nhất} \\
x \wedge x &= (x \wedge x) \vee 0 && \text{luật đồng nhất} \\
&= (x \wedge x) \vee (x \wedge \overline{x}) && \text{luật bù} \\
&= x \wedge (x \vee \overline{x}) && \text{luật phân phối} \\
&= x \wedge 1 && \text{luật bù} \\
&= x && \text{luật đồng nhất}
\end{aligned}
$$

Biểu diễn các 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* $^{-}$
- 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

- 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 $F(x, y, z) = x\overline{y}z$
- 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 $G(x, y, z) = xy\overline{z} + \overline{x}y\overline{z}$

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

Các biểu thức như trên được gọi là *khai triển tổng các tích (sum-of-products expansion)* hay *dạng tuyển chuẩn tắc (Disjunctive Normal Form - DNF)* của các hàm F và G

Biểu diễn các 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 $F(x, y, z) = (x + y)\overline{z}$

$$
\begin{aligned}
(x+y)\overline{z} &= x\overline{z} + y\overline{z} && \text{luật phân phối} \\
&= x1\overline{z} + 1y\overline{z} && \text{luật đồng nhất} \\
&= x(y+\overline{y})\overline{z} + (x+\overline{x})y\overline{z} && \text{tính chất đơn vị} \\
&= xy\overline{z} + x\overline{y}\overline{z} + xy\overline{z} + \overline{x}y\overline{z} && \text{luật phân phối} \\
&= xy\overline{z} + x\overline{y}\overline{z} + \overline{x}y\overline{z} && \text{luật lũy đẳng}
\end{aligned}
$$

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

Biểu diễn các 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 $+, \cdot, \overline{}$. Tập các toán tử $\{+, \cdot, \overline{}\}$ 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 $\{+, \overline{}\}$ là một tập đầy đủ (luật De Morgan)
  - Tập $\{\cdot, \overline{}\}$ là một tập đầy đủ (luật De Morgan)
  - Tập $\{+, \cdot\}$ không là một tập đầy đủ (ví dụ, ta không thể biểu diễn $F(x) = \overline{x}$ 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

Ta định nghĩa hai toán tử $|$ hay NAND và $\downarrow$ hay NOR

x	y	$x \mid y$ (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 6

Chứng minh rằng

(a) $\overline{x} = x \mid x$

(b) $xy = (x \mid y) \mid (x \mid y)$

(c) $x + y = (x \mid x) \mid (y \mid y)$

Bài tập 7

Chứng minh rằng

(a) $\overline{x} = x \downarrow x$

(b) $xy = (x \downarrow x) \downarrow (y \downarrow y)$

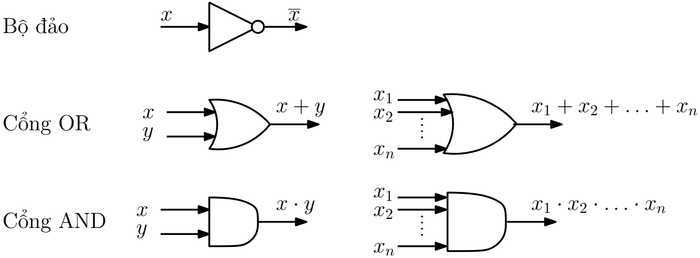
(c) $x + y = (x \downarrow y) \downarrow (x \downarrow y)$

Các cổng lôgic

- 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 lôgic

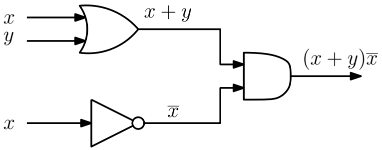
- *Bộ đảo (inverter):* đầu vào là giá trị của một biến Boole và đầu ra là phần bù của giá trị đầu vào
- *Cổng OR (OR gate):* đầu vào là giá trị của hai hay nhiều biến Boole và đầu ra là tổng Boole của các giá trị đó
- *Cổng AND (AND gate):* đầu vào là giá trị của hai hay nhiều biến Boole và đầu ra là tích Boole của các giá trị đó



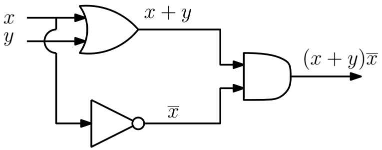
Ví dụ 7

Mạch tổ hợp có đầu ra là $(x+y)\overline{x}$

Cách 1



Cách 2



Bài tập 8

Vẽ các mạch tổ hợp có đầu ra

(a) $\overline{\overline{x}(y+\overline{z})}$ (b) $(x+y+z)(\overline{x}+\overline{y}+\overline{z})$

Các cổng lôgic

Ví dụ 8

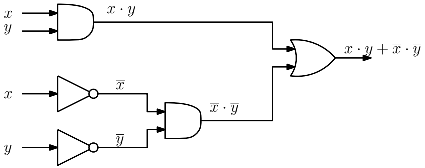
Đôi khi một hệ thống đèn cố định được điều khiển bởi hai hay nhiều công tắc. Các mạch điện cần được thiết kế sao cho khi ấn một công tắc bất kỳ thì hệ thống đèn đang bật sẽ tắt và hệ thống đèn đang tắt sẽ bật. Ta thiết kế một mạch thực hiện điều đó khi có hai công tắc

- 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 $xy + \overline{x}\overline{y}$

Các cổng lôgic

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 9

Hãy thử thiết kế mạch theo yêu cầu trong Ví dụ 8 với ba công tắc

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 lôgic 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 9

- 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 $xyz + x\overline{y}z$. Để biểu diễn mạch này, ta cần ba cổng và một bộ đảo
- Mặt khác, ta có thể viết $xyz + x\overline{y}z = xz(y + \overline{y}) = xz$. Lúc này, ta có thể biểu diễn mạch chỉ với một cổng

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

$$x \cdot y + x \cdot \overline{y} = x \cdot (y + \overline{y}) = x$$

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	$\overline{y}$
x	$x \cdot y$	$x \cdot \overline{y}$
$\overline{x}$	$\overline{x} \cdot y$	$\overline{x} \cdot \overline{y}$

Rút gọn hàm Boole

Ví dụ 10

Rút gọn các khai triển tổng các tích sau bằng sơ đồ Karnaugh

(a) $xy + \overline{x}y$

(b) $x\overline{y} + \overline{x}y$

(c) $x\overline{y} + \overline{x}y + \overline{x}\,\overline{y}$

	y	$\overline{y}$
x	1	
$\overline{x}$	1	

(a) $xy + \overline{x}y$

$\Downarrow$

y

	y	$\overline{y}$
x		1
$\overline{x}$	1	

(b) $x\overline{y} + \overline{x}y$

	y	$\overline{y}$
x		1
$\overline{x}$	1	1

(c) $x\overline{y} + \overline{x}y + \overline{x}\,\overline{y}$

$\Downarrow$

$\overline{x} + \overline{y}$

Rút gọn hàm Boole

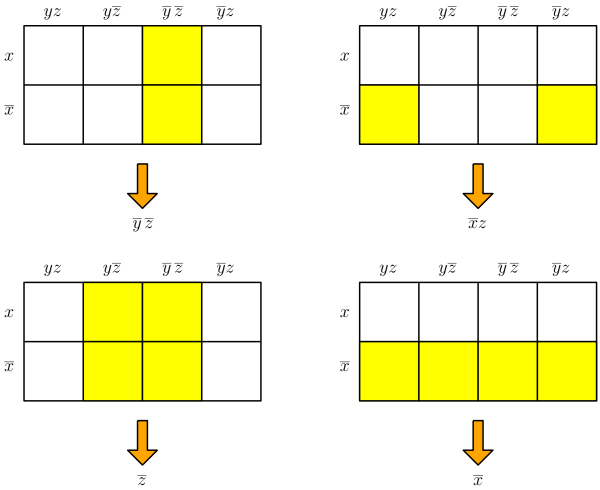
Sơ đồ Karnaugh cho hàm Boole có 3 biến

- 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 tục biến
- Các tiểu hạng của 2 ô kề nhau có số 1 có thể tổ hợp thành tích của hai tục biến
- Các tiểu hạng của 4 ô kề nhau có số 1 có thể tổ hợp thành một tục 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	$y\overline{z}$	$\overline{y}\,\overline{z}$	$\overline{y}z$
x	xyz	$xy\overline{z}$	$x\overline{y}\,\overline{z}$	$x\overline{y}z$
$\overline{x}$	$\overline{x}yz$	$\overline{x}y\overline{z}$	$\overline{x}\,\overline{y}\,\overline{z}$	$\overline{x}\,\overline{y}z$

Rút gọn hàm Boole

Ví dụ 11



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

Ví dụ 12

Rút gọn $xyz + x\overline{y}z + \overline{x}yz + \overline{x}\,\overline{y}z + \overline{x}\,\overline{y}\,\overline{z}$ 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
$x\overline{y}z$	101	2
$\overline{x}yz$	011	2
$\overline{x}\,\overline{y}z$	001	1
$\overline{x}\,\overline{y}\,\overline{z}$	000	0

- Nhóm các chuỗi nhị phân theo số các số 1

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	$x\overline{y}z$	101	✓
3	$\overline{x}yz$	011	✓
4	$\overline{x}\,\overline{y}z$	001	✓
5	$\overline{x}\,\overline{y}\,\overline{z}$	000	✓

	Số hạng	Chuỗi bit	
$(1,2)$	xz	1-1	✓
$(1,3)$	yz	-11	✓
$(2,4)$	$\overline{y}z$	-01	✓
$(3,4)$	$\overline{x}z$	0-1	✓
$(4,5)$	$\overline{x}\,\overline{y}$	00-	

	Số hạng	Chuỗi bit
$(1,2,3,4)$	z	- -1
$(1,3,2,4)$	z	- -1

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 $\overline{x}\,\overline{y}$
  - Lập bảng kiểm tra xem các ứng viên z và $\overline{x}\,\overline{y}$ có thực sự phủ hết các tiểu hạng gốc của hàm $F(x, y, z)$ ban đầu hay không

	xyz	$x\overline{y}z$	$\overline{x}yz$	$\overline{x}\,\overline{y}z$	$\overline{x}\,\overline{y}\,\overline{z}$
z	X	X	X	X	
$\overline{x}\,\overline{y}$				X	X

- Do đó, kết quả cuối cùng là $F(x, y, z) = z + \overline{x}\,\overline{y}$

Rút gọn hàm Boole

Ví dụ 13

Sử dụng phương pháp Quine-McCluskey để rút gọn biểu diễn của $F(w,x,y,z)$ sau $wxy\overline{z} + w\overline{x}yz + w\overline{x}y\overline{z} + \overline{w}xyz + \overline{w}x\overline{y}z + \overline{w}\,\overline{x}yz + \overline{w}\,\overline{x}\,\overline{y}z$

	Số hạng	Chuỗi bit	
1	$wxy\overline{z}$	1110	✓
2	$w\overline{x}yz$	1011	✓
3	$\overline{w}xyz$	0111	✓
4	$w\overline{x}y\overline{z}$	1010	✓
5	$\overline{w}x\overline{y}z$	0101	✓
6	$\overline{w}\,\overline{x}yz$	0011	✓
7	$\overline{w}\,\overline{x}\,\overline{y}z$	0001	✓

	Số hạng	Chuỗi bit	
$(1,4)$	$wy\overline{z}$	1-10	
$(2,4)$	$w\overline{x}y$	101-	
$(2,6)$	$\overline{x}yz$	-011	
$(3,5)$	$\overline{w}xz$	01-1	✓
$(3,6)$	$\overline{w}yz$	0-11	✓
$(5,7)$	$\overline{w}\,\overline{y}z$	0-01	✓
$(6,7)$	$\overline{w}\,\overline{x}z$	00-1	✓

	Số hạng	Chuỗi bit
$(3,5,6,7)$	$\overline{w}z$	0- -1
$(3,6,5,7)$	$\overline{w}z$	0- -1

	$wxy\overline{z}$	$w\overline{x}yz$	$\overline{w}xyz$	$w\overline{x}y\overline{z}$	$\overline{w}x\overline{y}z$	$\overline{w}\,\overline{x}yz$	$\overline{w}\,\overline{x}\,\overline{y}z$
$\overline{w}z$			X		X	X	X
$wy\overline{z}$	X			X			
$w\overline{x}y$		X		X			
$\overline{x}yz$		X				X	

Do đó, $F(w,x,y,z) = \overline{w}z + wy\overline{z} + w\overline{x}y$ hoặc $F(w,x,y,z) = \overline{w}z + wy\overline{z} + \overline{x}yz$

Rút gọn hàm Boole

Bài tập 10

Dùng sơ đồ Karnaugh và phương pháp Quine-McCluskey để rút gọn các khai triển tổng các tích sau

(a) $xy\overline{z} + x\overline{y}\,\overline{z} + \overline{x}yz + \overline{x}\,\overline{y}\,\overline{z}$

(b) $x\overline{y}z + x\overline{y}\,\overline{z} + \overline{x}yz + \overline{x}\,\overline{y}z + \overline{x}\,\overline{y}\,\overline{z}$

(c) $xyz + xy\overline{z} + x\overline{y}z + x\overline{y}\,\overline{z} + \overline{x}yz + \overline{x}\,\overline{y}z + \overline{x}\,\overline{y}\,\overline{z}$