

Đề bài dự án 10 tuần thay thế thực tập: Webapp tương tác giải thích cơ chế hoạt động của solver cho Combinatorial Reconfiguration

(Chọn solver: PARIS — planning-based solver cho Independent Set
Reconfiguration dưới token jump)

Đề xuất cho nhóm 3–5 sinh viên

Ngày 4 tháng 3 năm 2026

Tóm tắt nội dung

Mục tiêu của dự án là xây dựng một webapp mang tính *giải thích* (explanatory/educational) nhằm giúp người dùng hiểu rõ cơ chế hoạt động và các thuật toán cốt lõi của một solver cho bài toán *combinatorial reconfiguration*, tập trung vào *Independent Set Reconfiguration (ISR)* dưới quy tắc *token jump* theo chuẩn CoRe Challenge 2022 ([CoRe Challenge 2022 Website, 2022](#)). Dự án sử dụng PARIS (PARIS), một nhóm solver tiếp cận ISR như một bài toán *classical planning*: mã hoá instance thành bài toán PDDL/SAS+ và áp dụng các kỹ thuật lập kế hoạch hiện đại (heuristic search, symbolic search dựa trên BDD, và kỹ thuật phát hiện vô nghiệm bằng counter abstraction), có cả cấu hình *portfolio* ([Christen et al., 2023](#)). Mã và dữ liệu thực nghiệm của PARIS được cung cấp công khai ([PARIS Zenodo, 2023](#)).

Webapp cần mô tả và trực quan hoá (i) cách mô hình hoá ISR như một hệ chuyển trạng thái, (ii) cách PARIS biến đổi input ISR sang PDDL/SAS+ và ý nghĩa của biến/hành động/tiền điều kiện/tác động, (iii) cách các thuật toán lập kế hoạch tìm được (hoặc chứng minh không tồn tại) chuỗi reconfiguration, và (iv) cách đọc/diễn giải output và thống kê thực thi. Giao diện webapp bắt buộc bằng tiếng Anh, bắt buộc chạy PARIS *thực* ở backend, và phải tối ưu chi phí vận hành (ưu tiên free-tier/OSS).

Mục lục

1	Bối cảnh và lựa chọn solver	2
1.1	Combinatorial reconfiguration là gì?	2
1.2	Vì sao chọn PARIS?	2
2	Mục tiêu dự án và phạm vi	3
2.1	Mục tiêu tổng quát	3
2.2	Trách nhiệm tự chủ của nhóm	3
2.3	Ràng buộc bắt buộc (mandatory constraints)	3

2.4	Đối tượng người dùng (personas)	4
2.5	Phạm vi tối thiểu (MVP) và phạm vi mở rộng	4
3	Yêu cầu chức năng (Functional requirements)	5
3.1	Nhập liệu và quản lý instance	5
3.2	Trực quan hoá cấu hình và chuỗi reconfiguration	5
3.3	Giải thích cách PARIS giải ISR bằng lập kế hoạch (PDDL/SAS+)	5
3.4	Giải thích dữ liệu đầu vào/đầu ra	6
3.5	Chạy solver thực (bắt buộc)	6
4	Yêu cầu phi chức năng (Non-functional requirements)	7
5	Tiêu chí nghiệm thu và đánh giá (3 đầu điểm)	7
5.1	Thường xuyên: Báo cáo hàng tuần qua Google Sheet	7
5.2	Giữa kỳ: Implementation Plan Report + Algorithm Mechanism Report (PARIS)	8
5.3	Cuối kỳ: Nghiệm thu sản phẩm hoàn chỉnh (còn lại)	8
5.3.1	Tiêu chí bắt buộc (pass/fail)	8
5.3.2	Tiêu chí chấm điểm (tham khảo, 70%)	9
5.4	Mốc nghiệm thu và nộp bài	9
5.5	Danh mục kiểm tra nghiệm thu cuối kỳ (checklist)	9
6	Sản phẩm bàn giao (Deliverables)	10
6.1	Yêu cầu bắt buộc về thông tin nhóm	10
7	Tài liệu tham khảo	11

1 Bối cảnh và lựa chọn solver

1.1 Combinatorial reconfiguration là gì?

Trong một bài toán reconfiguration, ta có:

- Một bài toán tổ hợp gốc (ví dụ: independent set, vertex cover, colouring, Hamiltonian cycle, ...).
- Hai nghiệm khả thi S_{start} và S_{goal} .
- Một quy tắc “kề nhau” (adjacency rule) mô tả một bước biến đổi hợp lệ (ví dụ: thay đổi đúng 1 phần tử, token jumping/sliding, ...).

Nhiệm vụ điển hình: quyết định có tồn tại chuỗi biến đổi hợp lệ từ S_{start} đến S_{goal} hay không, hoặc tìm chuỗi ngắn nhất/dài nhất theo một tiêu chí.

1.2 Vì sao chọn PARIS?

Trang tổng hợp phần mềm của cộng đồng reconfiguration liệt kê PARIS (PARIS) như một solver cho ISR dựa trên kỹ thuật *planning* (Reconf Wiki Softwares, 2024). Bài báo ECAI 2023 mô tả chi tiết cách mô hình hoá ISR thành bài toán lập kế hoạch cổ điển (PDDL/SAS+) và các thành phần giải (heuristic search với landmarks, symbolic search

dựa trên BDD, và counter abstraction để phát hiện vô nghiệm), cũng như cấu hình *portfolio* cho nhiều track (Christen et al., 2023). Ngoài ra, nhóm tác giả công bố mã nguồn và dữ liệu thực nghiệm qua Zenodo (PARIS Zenodo, 2023). Những yếu tố này giúp dự án có:

- Nền tảng học thuật rõ ràng (bài báo hội nghị có phản biện).
- Câu chuyện “giải thích cơ chế” rất phù hợp cho webapp: người dùng thấy được chuỗi chuyển đổi $ISR \rightarrow \text{planning encoding} \rightarrow \text{planner search} \rightarrow \text{plan} \rightarrow \text{sequence}$.
- Dễ thiết kế demo tương tác: trực quan hoá state, action applicability, heuristic guidance, và chiến lược portfolio/timeout.

2 Mục tiêu dự án và phạm vi

2.1 Mục tiêu tổng quát

Xây dựng webapp có khả năng:

1. **Giải thích:** người dùng hiểu được các thành phần của solver PARIS và logic của thuật toán planning-based reconfiguration.
2. **Trực quan hoá:** minh hoạ bằng hình/animation/step-by-step cho các ví dụ nhỏ.
3. **Tương tác:** cho phép người dùng thay đổi tham số (bound, rule, instance nhỏ, ...) và quan sát ảnh hưởng lên quá trình giải.
4. **Tái sử dụng:** có tài liệu, mã nguồn, và cấu trúc đủ chuẩn để người khác mở rộng sang solver khác hoặc bài toán khác.

2.2 Trách nhiệm tự chủ của nhóm

Nhóm sinh viên **tự chịu trách nhiệm** (i) thiết kế kiến trúc hệ thống và lựa chọn công nghệ, (ii) lập kế hoạch thực hiện trong 10 tuần, và (iii) phân công nhiệm vụ, quy ước quy trình làm việc (issue/PR/review). Đề bài **không áp đặt** kiến trúc hay lịch trình chi tiết; các lựa chọn của nhóm sẽ được đánh giá thông qua sản phẩm bàn giao và nghiệm thu (xem Mục 5).

2.3 Ràng buộc bắt buộc (mandatory constraints)

- **Webapp phải bằng tiếng Anh:** toàn bộ nội dung hướng tới người dùng trong webapp (UI labels, hướng dẫn, tooltip, thông báo lỗi, trang giải thích trong app) phải bằng tiếng Anh.
- **Bắt buộc có chạy solver thực:** webapp phải thực sự gọi và chạy solver PARIS (executable hoặc Docker container) trên backend cho ít nhất các instance nhỏ; không được chỉ “trình diễn” bằng trace dựng sẵn.
- **Tiết kiệm chi phí hết mức có thể:** ưu tiên hạ tầng miễn phí/free-tier và phần mềm mã nguồn mở; tránh dịch vụ trả phí hoặc API trả tiền. Nếu bắt buộc dùng thành phần trả phí, phải nêu rõ lý do và chứng minh mức chi phí định kỳ là tối thiểu (mục tiêu: gần 0, lý tưởng là trong free-tier).

- **Nhập đồ thị & start/goal qua UI (click/drag/drop) — bắt buộc.**
- **Front-end bắt buộc dùng GitHub Pages:** trang web phía người dùng (front-end) phải được build và deploy lên GitHub Pages (có thể dùng GitHub Actions để tự động deploy). Backend (nếu cần để chạy solver) được phép triển khai ở nơi khác miễn là chi phí tối thiểu.

2.4 Đối tượng người dùng (personas)

- **Sinh viên/nhà nghiên cứu mới:** muốn hiểu mô hình hoá reconfiguration (state graph, move model) và ý tưởng thuật toán dựa trên planning.
- **Người dùng solver:** có instance nhỏ và muốn hiểu vì sao solver kết luận reachable/unreachable, vì sao chạy lâu, và ý nghĩa các thống kê (kích thước biểu diễn, thời gian, v.v.).
- **Giảng viên:** muốn một công cụ minh hoạ cho bài giảng về combinatorial reconfiguration và decision diagrams (planning/BDD).

2.5 Phạm vi tối thiểu (MVP) và phạm vi mở rộng

MVP (bắt buộc).

- Tập trung vào **Independent Set Reconfiguration (ISR)** dưới mô hình **token jump** theo chuẩn **CoRe Challenge 2022** ([CoRe Challenge 2022 Website, 2022](#)).
- Nhập/xem instance nhỏ: đồ thị G và nghiệm start/goal s, t thông qua (i) UI editor (click/drag/drop) và/hoặc (ii) import từ định dạng `*.col/*.dat` ([CoRe Challenge 2022 Website, 2022](#)).
- Trực quan hoá chuỗi reconfiguration và từng bước chuyển (sequence viewer) *từ kết quả chạy thật của PARIS* (FR-15).
- Trang “giải thích thuật toán”: giải thích (i) state graph của ISR, (ii) mã hoá ISR sang PDDL/SAS+, và (iii) cách planner tìm plan/chuỗi (hoặc phát hiện vô nghiệm) ([Christen et al., 2023](#)).

Mở rộng (tuỳ thời gian).

- Cho phép người dùng chọn *chế độ giải* tương ứng với các thành phần trong PARIS: (i) GBFS+Landmarks, (ii) A*+Landmarks (hướng tới tối ưu), (iii) symbolic search (BDD-based), (iv) counter abstraction (phát hiện vô nghiệm), và (v) portfolio (chạy tuần tự nhiều thành phần với timeout) ([Christen et al., 2023](#)).
- Chế độ “explain-by-contrast”: chạy 2 cấu hình khác nhau trên cùng instance và so sánh số node mở rộng, heuristic, hoặc thời gian.
- Tích hợp kiểm chứng output bằng validator (hoặc kiểm chứng nội bộ) và cho phép tải file `*.out` chuẩn ([CoRe Challenge 2022 Website, 2022](#)).
- So sánh với solver khác (ví dụ: ASP-based) dựa trên thông tin tổng quan ở trang Softwares ([Reconf Wiki Softwares, 2024](#)).

3 Yêu cầu chức năng (Functional requirements)

Trong phần này, mỗi yêu cầu có mã định danh FR-# để phục vụ theo dõi.

3.1 Nhập liệu và quản lý instance

FR-01: Thư viện ví dụ (Example Library)

Webapp cung cấp ít nhất 8–12 ví dụ “nhỏ” ($n \leq 12$ đỉnh) với mô tả ngắn gọn, bao gồm cả ví dụ reachable và unreachable.

FR-02: Nhập đồ thị qua UI (click/drag/drop) — bắt buộc

Người dùng có thể tạo và chỉnh sửa đồ thị trực tiếp trên UI bằng thao tác click và kéo/thả: (i) thêm/xoá/di chuyển đỉnh, (ii) thêm/xoá cạnh, (iii) undo/redo tối thiểu cho các thao tác chỉnh sửa cơ bản.

FR-03: Nhập tập start/goal qua UI — bắt buộc

Người dùng có thể chọn S_{start} và S_{goal} trực tiếp bằng cách click lên các đỉnh (toggle token), có hiển thị rõ kích thước, tính hợp lệ, và khác biệt giữa start/goal.

FR-04: Import/export định dạng solver (khuyến nghị)

Webapp cho phép import/export (hoặc copy/paste) instance theo định dạng mà PARIS chấp nhận (DIMACS + file mô tả s, t), nhằm đảm bảo tương thích và tái hiện kết quả (CoRe Challenge 2022 Website, 2022; PARIS Zenodo, 2023).

FR-05: Kiểm tra hợp lệ

Hệ thống tự kiểm tra start/goal có hợp lệ không (ví dụ: independent set, kích thước đúng, ...) và báo lỗi bằng tiếng Anh.

3.2 Trực quan hoá cấu hình và chuỗi reconfiguration

FR-06: Graph viewer

Hiển thị đồ thị G (layout tự động), tô màu token ở mỗi thời điểm t .

FR-07: Sequence player

Có thanh timeline (step $0..L$), nút play/pause, next/prev; mỗi bước hiển thị phép chuyển (đỉnh rời/đỉnh vào) và giải thích vì sao bước đó hợp lệ theo rule.

FR-08: Constraint overlay

Cho phép bật/tắt overlay để minh hoạ constraint (ví dụ: cấm adjacency trong independent set), và highlight các vi phạm khi người dùng chọn start/goal không hợp lệ.

3.3 Giải thích cách PARIS giải ISR bằng lập kế hoạch (PDDL/SAS+)

FR-09: Giải thích mô hình trạng thái

Webapp phải giải thích rõ “trạng thái” là một independent set hợp lệ và “một bước” là phép *token jump* (di chuyển đúng 1 token sang một đỉnh trống) sao cho trạng thái mới vẫn là independent set (CoRe Challenge 2022 Website, 2022).

FR-10: Giải thích bước mã hoá (encoding) sang planning

Webapp phải trình bày và minh hoạ cách PARIS mã hoá instance ISR thành bài toán planning: (i) biến trạng thái (ví dụ: token-at/occupied, hoặc biến đếm theo token), (ii) hành động tương ứng với một bước token jump, (iii) tiên điều kiện đảm bảo tính độc lập (không tạo adjacency), và (iv) điều kiện đích tương ứng với I_t . Phần này phải có ví dụ nhỏ ($n \leq 10$) kèm minh hoạ trực quan về “một action áp dụng được” và “vì sao action bị cấm” (Christen et al., 2023).

FR-11: Giải thích thuật toán tìm lời giải/không lời giải

Webapp phải giải thích ở mức khái niệm (high-level) ít nhất các ý sau: (i) heuristic search (ví dụ A* hoặc GBFS) và vai trò của *landmarks* trong hướng dẫn tìm kiếm, (ii) symbolic search dùng BDD để biểu diễn tập trạng thái, (iii) ý tưởng counter abstraction để phát hiện vô nghiệm, và (iv) cơ chế *portfolio* (chạy nhiều thành phần với timeout khác nhau) trong PARIS (Christen et al., 2023). Mỗi ý phải có minh hoạ trực quan tối thiểu (ví dụ: frontier/expanded nodes, heuristic values, hoặc minh hoạ tập trạng thái dạng symbolic).

3.4 Giải thích dữ liệu đầu vào/đầu ra

FR-12: Hướng dẫn định dạng input

Webapp phải hướng dẫn (kèm ví dụ) định dạng input ISR theo chuẩn CoRe Challenge 2022: cặp file *.col (DIMACS cho đồ thị) và *.dat (dòng s và t cho start/target) (CoRe Challenge 2022 Website, 2022). Webapp phải hỗ trợ import/export (hoặc copy/paste) các định dạng này.

FR-13: Giải thích output và kiểm chứng chuỗi

Webapp phải giải thích output *.out (các dòng a YES/NO và chuỗi trạng thái), và có cơ chế kiểm chứng nội bộ tối thiểu: mỗi bước chỉ thay đổi đúng 1 đỉnh và trạng thái luôn là independent set (CoRe Challenge 2022 Website, 2022).

FR-14: Glossary

Danh mục thuật ngữ: independent set, token jump, state graph, planning task, PDDL/SAS+, heuristic search, landmarks, symbolic search, BDD, portfolio, v.v.

3.5 Chạy solver thực (bắt buộc)

FR-15: Chạy PARIS trên backend (bắt buộc)

Backend phải chạy PARIS thật sự (không dùng dữ liệu dựng sẵn). Webapp phải lấy được (i) log thực thi (thông tin tổng quát, thời gian, trạng thái thành công/thất bại), và (ii) chuỗi reconfiguration (nếu tìm được) để hiển thị trong sequence player (Christen et al., 2023; PARIS Zenodo, 2023).

FR-16: Giới hạn tài nguyên

Đặt timeout và giới hạn kích thước instance để tránh treo hệ thống; có thông báo rõ ràng khi vượt giới hạn. Khuyến nghị chỉ cho phép instance nhỏ/trung bình (ví dụ $n \leq 50$) cho chế độ chạy trực tuyến, và cung cấp chế độ “offline run” (tải file output) nếu nhóm muốn hỗ trợ instance lớn.

FR-17: Cache kết quả

Cache theo hash của input (.col+.dat+option) để giảm tải và giảm chi phí chạy.

4 Yêu cầu phi chức năng (Non-functional requirements)

NFR-01: Tính đúng đắn của nội dung

Các giải thích thuật toán phải bám sát tài liệu chính thức (README) và/hoặc bài báo có phản biện (ưu tiên (CoRe Challenge 2022 Website, 2022; PARIS Zenodo, 2023)).

NFR-02: Trải nghiệm người dùng

UI rõ ràng; người dùng mới có thể chạy được một ví dụ trong <2 phút.

NFR-03: Khả năng tái hiện (reproducibility)

Có README mô tả cài đặt, chạy local bằng Docker, và quy trình build/deploy.

NFR-04: Bảo mật cơ bản

Nếu có backend chạy solver: chống upload file tùy ý; validate input; giới hạn thời gian chạy.

NFR-05: Khả năng mở rộng nội dung

Đễ thêm ví dụ mới, thêm rule mới, hoặc chuyển sang solver khác.

NFR-06: Ngôn ngữ

Tất cả nội dung trong webapp hướng tới người dùng phải bằng tiếng Anh (không yêu cầu mã nguồn/ghi chú nội bộ bằng tiếng Anh).

NFR-07: Chi phí

Giải pháp triển khai/vận hành phải tối ưu chi phí: ưu tiên chạy được hoàn toàn trên free-tier hoặc chi phí gần 0; phải có cơ chế giới hạn tài nguyên (timeout, quota) để tránh phát sinh chi phí và tránh lạm dụng.

NFR-08: Front-end trên GitHub Pages (bắt buộc)

Front-end phải được triển khai bằng GitHub Pages; quy trình build/deploy nên tự động hoá bằng GitHub Actions để tránh thao tác thủ công.

5 Tiêu chí nghiệm thu và đánh giá (3 đầu điểm)

Phần nghiệm thu/đánh giá được chia thành 3 đầu điểm tương ứng: **thường xuyên** (báo cáo hàng tuần), **giữa kỳ** (2 báo cáo bắt buộc), và **cuối kỳ** (các yêu cầu còn lại + sản phẩm hoàn chỉnh).

5.1 Thường xuyên: Báo cáo hàng tuần qua Google Sheet

Pass/Fail (bắt buộc).

- Điền báo cáo **mỗi tuần** qua Google Sheet theo mẫu thực tập.
- Link Google Sheet phải được đính kèm trong báo cáo kỹ thuật và cung cấp tại buổi nghiệm thu cuối kỳ.

Chấm điểm (tham khảo, 10%). Chất lượng báo cáo tuần được chấm dựa trên: tính đều đặn, mức độ cụ thể (có minh chứng như link commit/PR), và sự minh bạch về đóng góp từng thành viên.

5.2 Giữa kỳ: Implementation Plan Report + Algorithm Mechanism Report (PARIS)

Yêu cầu nộp (bắt buộc, deadline: trong < 5 tuần đầu).

- **Implementation Plan Report** (5–10 trang): mô tả và giải thích rõ ràng về (i) cách thiết kế hệ thống mà nhóm chọn, (ii) kế hoạch thực hiện trong 10 tuần (mốc theo tuần/sprint và deliverables), và (iii) phân công nhiệm vụ/cơ chế phối hợp.
- **Algorithm Mechanism Report (PARIS)** (8–15 trang): giải thích chi tiết cơ chế hoạt động của thuật toán trong PARIS, dựa trên tài liệu chính thức và bài báo ([CoRe Challenge 2022 Website, 2022](#); [PARIS Zenodo, 2023](#)); có minh họa/diagram và liên hệ tới log/output khi chạy thật.

Pass/Fail (bắt buộc). Cả hai báo cáo phải được nộp đúng hạn; nội dung phải nhất quán, có trích dẫn tài liệu; và phải tuân thủ yêu cầu thông tin nhóm (Mục 6.1).

Chấm điểm (tham khảo, 20%).

- Implementation Plan Report: 10%.
- Algorithm Mechanism Report: 10%.

5.3 Cuối kỳ: Nghiệm thu sản phẩm hoàn chỉnh (còn lại)

5.3.1 Tiêu chí bắt buộc (pass/fail)

1. **Front-end GitHub Pages:** webapp front-end phải được build và deploy lên GitHub Pages, có URL truy cập công khai/được cấp quyền.
2. Webapp chạy được (deploy hoặc chạy local), có ít nhất 8 ví dụ trong thư viện (FR-01).
3. **Nhập đồ thị và start/goal qua UI (bắt buộc):** người dùng nhập/chỉnh sửa đồ thị bằng click/drag/drop và chọn S_{start}, S_{goal} bằng thao tác trên UI (FR-02, FR-03), có kiểm tra hợp lệ (FR-05).
4. Có sequence player hiển thị chuỗi reconfiguration theo bước (FR-07) trên graph viewer (FR-06).
5. **Solver thực (bắt buộc):** từ webapp có thể chạy PARIS thật sự trên backend cho các instance nhỏ, lấy log và (nếu có) chuỗi kết quả để hiển thị (FR-15, FR-16).
6. **Giải thích thuật toán (bắt buộc):** có trang/luồng nội dung giải thích mô hình trạng thái, vai trò của planning, cách tìm đường đi ngắn nhất và truy vết dựng chuỗi, cùng glossary (FR-09–FR-14).
7. **Ngôn ngữ (bắt buộc):** toàn bộ nội dung hướng tới người dùng trong webapp phải bằng tiếng Anh (NFR-06).
8. **Chi phí (bắt buộc):** cung cấp minh chứng triển khai/vận hành chi phí tối thiểu (ưu tiên free-tier), kèm mô tả cơ chế giới hạn tài nguyên để kiểm soát chi phí (NFR-07).
9. **Thông tin nhóm (bắt buộc):** tất cả các tài liệu bàn giao (báo cáo, mã nguồn, slide) phải chứa đầy đủ thông tin các thành viên theo yêu cầu ở Mục 6.1.

5.3.2 Tiêu chí chấm điểm (tham khảo, 70%)

- **Nội dung thuật toán và tính đúng đắn (20%):** giải thích đúng mô hình trạng thái/move model; mô tả đúng pipeline planning-based; diễn giải đúng reachable/unreachable/shortest sequence.
- **Trực quan hoá và UX (25%):** graph editor/sequence player dễ dùng; trực quan; có chú thích, tooltip, và thông báo lỗi hữu ích (bằng tiếng Anh).
- **Chất lượng kỹ thuật (15%):** tích hợp PARIS ổn định, có kiểm soát tài nguyên, cấu trúc code sạch, có test/CI ở mức hợp lý.
- **Tài liệu và khả năng tái hiện (10%):** hướng dẫn chạy local/deploy rõ ràng; tái hiện được kết quả trên các ví dụ.
- **Tối ưu chi phí (0–5%):** lựa chọn hạ tầng gần 0 chi phí; có quota/timeout/cache; có ước lượng chi phí định kỳ và chứng minh là tối thiểu.

Ghi chú: tỷ trọng các mục có thể điều chỉnh nhỏ tùy giảng viên; tuy nhiên 3 đầu điểm (thường xuyên/giữa kỳ/cuối kỳ) và các mục pass/fail là bắt buộc.

5.4 Mốc nghiệm thu và nộp bài

Mốc	Yêu cầu nộp / trình diễn
Trong ≤ 5 tuần đầu	<i>Implementation Plan Report</i> + <i>Algorithm Mechanism Report</i> (bắt buộc, giữa kỳ).
Tuần 6–7	Trình diễn MVP: graph editor + chạy solver thật cho ví dụ nhỏ + sequence player + trang giải thích.
Tuần 10 (cuối kỳ)	Nghiệm thu cuối: toàn bộ deliverables + kiểm tra checklist + chấm rubric.

5.5 Danh mục kiểm tra nghiệm thu cuối kỳ (checklist)

1. Front-end deploy trên GitHub Pages và truy cập được.
2. Có hướng dẫn cài đặt/chạy và có thể tái lập kết quả với các ví dụ kèm theo.
3. Graph editor cho phép nhập đồ thị và start/goal bằng click/drag/drop; có kiểm tra hợp lệ và thông báo lỗi tiếng Anh.
4. Sequence player thể hiện rõ trạng thái hiện tại, bước kế tiếp, và phép chuyển hợp lệ.
5. Trang “giải thích solver”: mô tả pipeline *encode* \rightarrow *solve* \rightarrow *decode/trace* và cách đọc output, gắn với PARIS.
6. **Minh chứng solver chạy thật:** ít nhất 3 instance do nhóm chọn (log/video/ảnh chụp).
7. **Báo cáo giữa kỳ (≤ 5 tuần):** *Implementation Plan Report* và *Algorithm Mechanism Report* đầy đủ.

8. **Google Sheet báo cáo hàng tuần:** đủ 15 lần cập nhật, có link trong báo cáo kỹ thuật.
9. Tài liệu và mã nguồn đủ rõ để một nhóm khác có thể thêm tối thiểu 1 ví dụ mới hoặc 1 phần hiển thị mới trong ≤ 2 giờ.
10. Mọi tài liệu bàn giao (báo cáo, slide) và mã nguồn đều có mục/đoạn thể hiện đầy đủ thông tin nhóm (Mục 6.1).

6 Sản phẩm bàn giao (Deliverables)

- Tập hợp tất cả tài liệu và sản phẩm vào một github repo duy nhất (có thể private nếu cần, nhưng phải cấp quyền truy cập cho giảng viên). Nếu sản phẩm nào không để được trong repo do kích thước hoặc lý do khác, phải có hướng dẫn rõ ràng trong repo để truy cập (ví dụ: link Google Drive, hướng dẫn build/run để tạo ra sản phẩm đó, v.v.).
- Mã nguồn (Git repo) + license.
- **Front-end trên GitHub Pages (bắt buộc):** URL GitHub Pages cho webapp phía người dùng.
- Backend (nếu tách riêng) + hướng dẫn triển khai chạy solver end-to-end; ưu tiên free-tier.
- Bộ ví dụ (ít nhất 8 instance nhỏ) kèm mô tả và dữ liệu để tái hiện.
- **Implementation Plan Report** (bắt buộc, nộp trong ≤ 5 tuần đầu; 5–10 trang).
- **Algorithm Mechanism Report (PARIS)** (bắt buộc, nộp trong ≤ 5 tuần đầu; 8–15 trang).
- **Báo cáo hàng tuần qua Google Sheet (bắt buộc):** điền báo cáo đều đặn mỗi tuần, có link trong báo cáo kỹ thuật.
- Báo cáo kỹ thuật (10–25 trang): mô tả thiết kế và hiện thực *as-built* (những gì đã làm), tích hợp PARIS, mô hình hoá/encoding, các quyết định kỹ thuật, đánh đổi, và hướng dẫn mở rộng.
- Slide thuyết trình + video demo (3–7 phút) thể hiện: nhập đề thị bằng UI, chạy solver thật, và xem chuỗi kết quả.
- Bản mô tả triển khai tối ưu chi phí (1–2 trang): lựa chọn hạ tầng, ước lượng chi phí định kỳ, và cơ chế kiểm soát tài nguyên (timeout/quota/cache).
- Minh chứng “solver chạy thật”: log/ảnh chụp/video cho ít nhất 3 instance do nhóm chọn, thể hiện backend thực sự gọi PARIS và trả về kết quả.

6.1 Yêu cầu bắt buộc về thông tin nhóm

Tất cả các báo cáo, mã nguồn, và slide thuyết trình phải có **đầy đủ thông tin các thành viên trong nhóm**. Tối thiểu gồm:

- Họ và tên; mã số sinh viên (nếu có); email; đơn vị/khoa/trường.

- Vai trò chính trong dự án (ví dụ: backend/solver integration, UI/visualization, documentation, DevOps).

Gợi ý vị trí đặt thông tin (không bắt buộc theo đúng mẫu):

- Báo cáo: trang bìa hoặc trang đầu.
- Slide: trang tiêu đề hoặc trang cuối.
- Mã nguồn: README và một file AUTHORS.md hoặc TEAM.md trong repo.

7 Tài liệu tham khảo

Tài liệu

Remo Christen, Salomé Eriksson, Michael Katz, Christian Muise, Alice Petrov, Florian Pommerening, Jendrik Seipp, Silvan Sievers, and David Speck. PARIS: Planning Algorithms for Reconfiguring Independent Sets. In *Proceedings of the 26th European Conference on Artificial Intelligence (ECAI 2023)*, pp. 453–460, 2023. PDF: <https://ai.dmi.unibas.ch/papers/christen-et-al-ecai2023.pdf>. Accessed 2026-03-02.

Christen, Remo and collaborators. Code and experimental data for the ECAI 2023 paper “PARIS: Planning Algorithms for Reconfiguring Independent Sets” (Zenodo dataset). <https://zenodo.org/records/8178793/latest>. Accessed 2026-03-02.

CoRe Challenge 2022 Organizers. CoRe Challenge 2022 website: problem definition and file format for ISR (token jump), including *.col, *.dat, and *.out. <https://core-challenge.github.io/2022/>. Accessed 2026-03-02.

Combinatorial Reconfiguration Wikidot. Softwares page (list of reconfiguration-related tools including PARIS). <https://reconf.wikidot.com/software>. Last edited 2024-10-18.

Malte Helmert. The Fast Downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.