[HCMUS] [IntroToAI] [Lab 1] - Các thuật toán tìm kiếm trên đồ thị

1. Thông tin chung

1.1. Thông tin liên hệ

- GVTH: Nguyễn Bảo Long
- Liên hệ: baolongnguyen.mac@gmail.com

1.2. Yêu cầu bài nộp

- Lab cá nhân.
- Sử dụng ngôn ngữ Python, phiên bản `3.9.12` với các thư viện được đính kèm trong `requirements.txt` (tạo môi trường Python bằng conda (recommended) hoặc venv).
- Nộp tệp nén có dạng `<MSSV>.zip`. Bên trong chứa 1 file `*.txt` chứa đường link video demo, 1 báo cáo dạng `*.pdf` và một thư mục chứa source code.
- Báo cáo viết:
  - Nộp file `*.pdf`. Phải có trang bìa, mục lục, nội dung.
  - Cần tự đánh giá trên thang điểm 10 đối với từng yêu cầu sau đó tính trung bình để có được đánh giá tổng quan.
  - Có thể dùng bất kỳ phần mềm nào để viết báo cáo. Tuy nhiên, đối với các văn bản khoa học, nên viết bằng Latex.

2. Mục tiêu & Đánh giá

2.1. Thang điểm

- Tìm hiểu và trình bày được các thuật toán tìm kiếm trên đồ thị - 4đ.
- So sánh các thuật toán với nhau - 2đ.
- Cài đặt được các thuật toán kể trên - 3đ.
- Tìm hiểu thêm các thuật toán tìm kiếm ngoài yêu cầu tại Yêu cầu kỹ thuật (khi tìm hiểu, phải đảm bảo được 3 yếu tố nêu trên) - 1đ.

2.2. Lưu ý

- Trích dẫn nguồn đầy đủ trong báo cáo.
- Không sao chép bài.
- Vi phạm 1 trong 2 lưu ý trên → 0đ.

3. Yêu cầu kỹ thuật

- Tương ứng với 3 mục tiêu chính trong phần Thang điểm, sẽ có 3 yêu cầu tương đương.

3.1. Tìm hiểu & Trình bày thuật toán

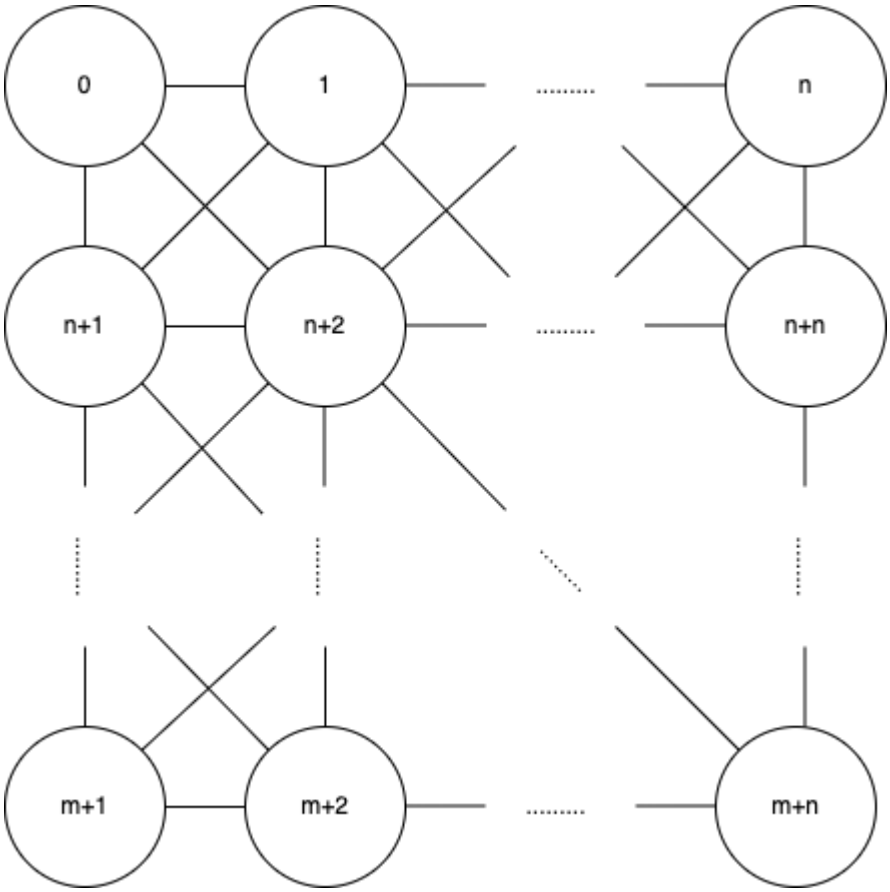
- Nêu các thành phần của 1 bài toán tìm kiếm và cách giải một bài toán tìm kiếm nói chung. Phân loại bài toán tìm kiếm (Uninformed Search và Informed Search).
- Trình bày về 4 thuật toán `DFS, BFS, UCS, A*`. Nội dung gồm có:
  - Ý tưởng chung.
  - Mã giả.
  - Đưa ra đánh giá về thuật toán (tính đầy đủ, tính tối ưu, độ phức tạp). Đối với `A*`, cần đề xuất heuristic và giải thích lý do tại sao chọn heuristic đó. Như thế nào là 1 heuristic chấp nhận được?
  - Một ví dụ đơn giản (đồ thị chứa 5-6 nodes) để minh họa cho mã giả. Tham khảo cách trình bày dạng cây trong video này. Lưu ý, ví dụ cho các thuật toán nên lấy cùng một đồ thị để thấy được ưu nhược điểm của từng thuật toán.

3.2. So sánh

- So sánh sự khác biệt giữa `UCS`, `Greedy` và `A*`.
- So sánh sự khác biệt giữa `UCS` và `Dijkstra`.

3.3. Cài đặt

- Nhiệm vụ là tìm kiếm đường đi giữa 2 node trong đồ thị như hình vẽ



3.3.1. Giải thích code

- File `Space.py` định nghĩa một không gian tìm kiếm. Không gian này là một đồ thị có dạng như hình vẽ và chứa `class Node`, `class Graph`:
  - `class Node`: Định nghĩa đối tượng node và các hàm bổ trợ.
  - `class Graph`: Định nghĩa đối tượng graph và các hàm bổ trợ.
- File `SearchAlgorithm.py`: Nơi cài đặt các thuật toán tìm kiếm. Các bạn có thể viết thêm các hàm bổ trợ nếu cảm thấy cần thiết. Tuy nhiên, không được thay đổi tham số của các hàm được yêu cầu hoàn thành. Một số thông tin sau có thể hữu ích cho bạn:
  - `open_set`: Tập mở, chứa các node được mở rộng đến.
  - `closed_set`: Tập đóng, chứa các mode đã được xét đến.
  - `father`: Từ node x mở rộng được node y thì `father[y] = x`.
  - `cost`: Chi phí đi từ trạng thái đầu đến node x là y thì `cost[x] = y`.
  - Các bạn có thể không sử dụng các gợi gợi ý về tập đóng, tập mở, chi phí.
- File `Constants.py`: Chứa các hằng số. Các bạn vui lòng không thay đổi giá trị các màu trong file này và có thể thay đổi tỷ lệ màn hình cho phù hợp với từng máy (số node vẫn phải giữ nguyên).
- File `main.py`: Các bạn gọi file này để thực thi chương trình. Ví dụ: `python main.py --algo AStar --start 71 --goal 318`.

3.3.2. Yêu cầu

- Các bạn đọc code được cung cấp, hoàn thành các hàm chưa được cài đặt: `DFS, BFS, UCS, AStar`.

- Với mỗi thuật toán, các bạn chụp màn hình lại kết quả cuối và mô tả ngắn gọn quá trình tìm kiếm vào báo cáo. Nêu nhận xét về kết quả thu được.
- Quá trình chạy phải được quay lại trong 1 video. Cần phân đoạn rõ ràng từng thuật toán. Upload video lên Youtube và đính kèm đường link vào file `*.txt`. Lưu ý, mỗi video không nên dài quá 5phút. Các bạn có thể tăng tốc video bằng phần mềm.
- Các bạn tham khảo video mẫu tại đường link này. Lưu ý **tuân thủ các quy định về màu sắc cho các loại node** (đọc file `Constants.py`).