BẢN BÁO CÁO

MÔN HỌC: CƠ SỞ TRÍ TUỆ NHÂN TẠO

ĐỒ ÁN: CÁC THUẬT TOÁN TÌM KIẾM

I. Thông tin nhóm

MSSV	Tên thành viên
20120524	Võ Đức Lợi
20120528	Nguyễn Thành Luân

II. Đánh giá tiến độ hoàn thành

Loại bản đồ	Mức độ	Tiến độ hoàn thành
Bản đồ không có điểm thưởng	Mức 1a	100%
	Mức 1b	100%
Bản đồ có điểm thưởng	Mức 2a	65% (chưa mô phỏng thuật toán trên bản đồ)
	Mức 2b	0% (chưa hoàn thành code cài đặt thuật toán)
Điểm cộng	Kịch bản nâng cấp	0%
	Video minh hoạ	0%

III. Giải thích và mô tả các thuật toán

Các đỉnh trong đồ thị thuộc class NODE có khác thuộc tính:

+ x – toạ độ x trong ma trận

+ y – toạ độ y trong ma trận

+ parent_Node – lưu trữ NODE cha để truy vết tìm kiếm (nếu có, mặc định là None)

+ bonus_Point – điểm thưởng (nếu có, mặc định = 0)

+ adjacency_List – danh sách các đỉnh kề (kiểu dữ liệu list, được lưu theo thứ tự đỉnh kề là trên, dưới, trái, phải với chi phí đều bằng 1)

1. Tìm kiếm không có thông tin

a) Thuật toán BFS

+ Bước 1: Tạo 1 hàng đợi (queue – kiểu dữ liệu list) chứa đỉnh nguồn và 1 danh sách các đỉnh đã xét (visited – kiểu dữ liệu list)

+ Bước 2: Nếu hàng đợi rỗng thì tìm kiếm thất bại, kết thúc thuật toán

+ Bước 3: Lấy đỉnh đầu trong hàng đợi ra và gọi nó là đỉnh v

+ Bước 4: Nếu v là đỉnh đích thì tìm kiếm thành công, kết thúc việc tìm kiếm

+ Bước 5: Duyệt qua tất cả các đỉnh kề với đỉnh v, đỉnh nào không thuộc queue và visited thì cho vào cuối queue, và quay trở lại bước 2

b) Thuật toán DFS

+ Bước 1: Tạo 1 ngăn xếp (stack – kiểu dữ liệu list) chứa đỉnh nguồn và 1 danh sách các đỉnh đã xét (visited – kiểu dữ liệu list)

+ Bước 2: Nếu ngăn xếp rỗng thì tìm kiếm thất bại, kết thúc thuật toán

+ Bước 3: Lấy đỉnh đầu trong hàng đợi ra và gọi nó là đỉnh v

+ Bước 4: Nếu v là đỉnh đích thì tìm kiếm thành công, kết thúc việc tìm kiếm

+ Bước 5: Duyệt qua tất cả các đỉnh kề với đỉnh v, cho tất cả các đỉnh mà không thuộc queue và visited vào đầu stack, và quay trở lại bước 2

c) Thuật toán UCS

+ Bước 1: Tạo 1 hàng đợi ưu tiên (priority_Queue – kiểu dữ liệu list) chứa đỉnh nguồn và 1 danh sách các đỉnh đã xét (visited – kiểu dữ liệu list)

+ Bước 2: Nếu hàng đợi rỗng thì tìm kiếm thất bại, kết thúc thuật toán

+ Bước 3: Lấy đỉnh đầu trong hàng đợi ra và gọi nó là đỉnh v

+ Bước 4: Nếu v là đỉnh đích thì tìm kiếm thành công, kết thúc việc tìm kiếm

+ Bước 5: Duyệt qua tất cả các đỉnh kề với đỉnh v, đỉnh nào không thuộc visited thì tính chi phí đường đi của đỉnh đó và kiểm tra xem đỉnh đó tồn tại trong priority_Queue hay chưa, nếu không tồn tại thì thêm vào priority_Queue kèm theo chi phí của đỉnh, nếu đã tồn tại thì xét: Nếu chi phí của đỉnh tồn tại trong priority_Queue nhỏ hơn hoặc bằng chi phí đỉnh đang xét thì bỏ qua đỉnh này, nếu chi phí của đỉnh tồn tại trong priority_Queue lớn hơn chi phí đỉnh đang xét thì xoá đỉnh trong priority_Queue và thêm đỉnh đang xét vào hàng đợi ưu tiên kèm theo chi phí. Quay lại bước 2

2. <u>Tìm kiếm có thông tin</u>

Ta định nghĩa 2 hàm heuristic:

+ Heuristic1: Đánh giá khoảng cách ước tính thông qua khoảng cách Euclid so với đỉnh đích

+ Heuristic2: Đánh giá khoảng cách ước tính thông qua khoảng cách Manhattan so với đỉnh đích

⇨ Cả 2 hàm heuristic đều đảm bảo tính nhất quán và hợp lí nên đường đi tìm được sẽ đảm bảo được tối ưu

a) Thuật toán GBFS

+ Các bước giống với thuật toán UCS nhưng thay vì đưa vào hàng đợi ưu tiên theo chi phí đường đi thì ta sử dụng khoảng cách ước tính để đưa vào hàng đợi ưu tiên

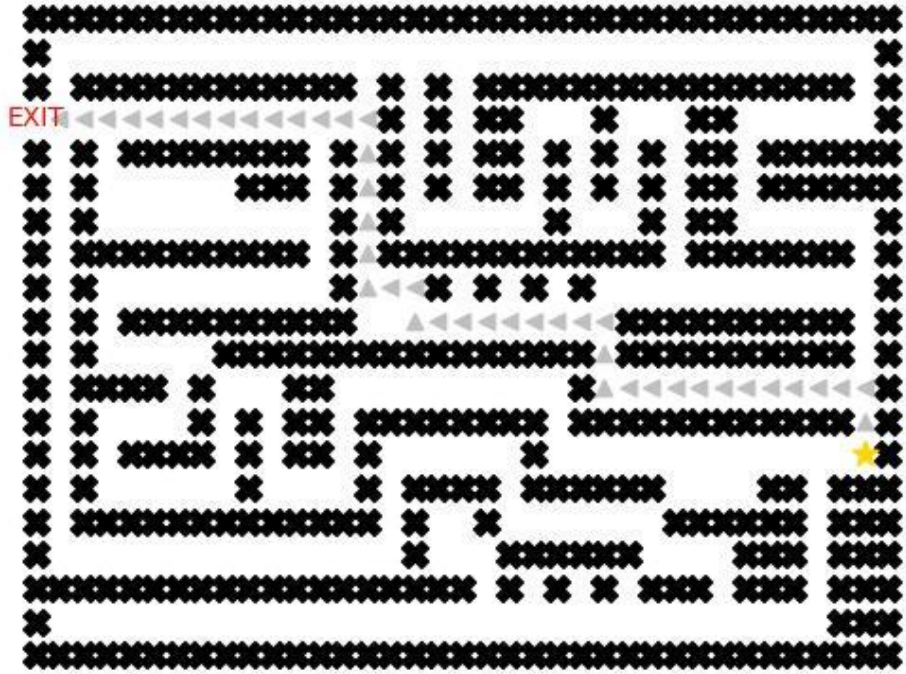
*b) Thuật toán A**

+ Các bước giống với thuật toán UCS nhưng thay vì đưa vào hàng đợi ưu tiên theo chi phí đường đi thì ta sử dụng chi phí ước tính = chi phí đường đi + khoảng cách ước tính để đưa vào hàng đợi ưu tiên

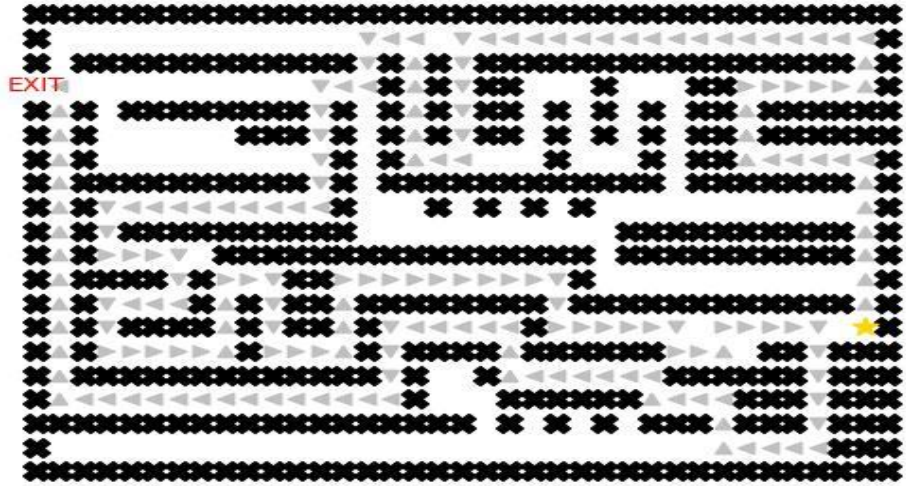
IV. Nhận xét các thuật toán qua các bản đồ

1. Bản đồ 1

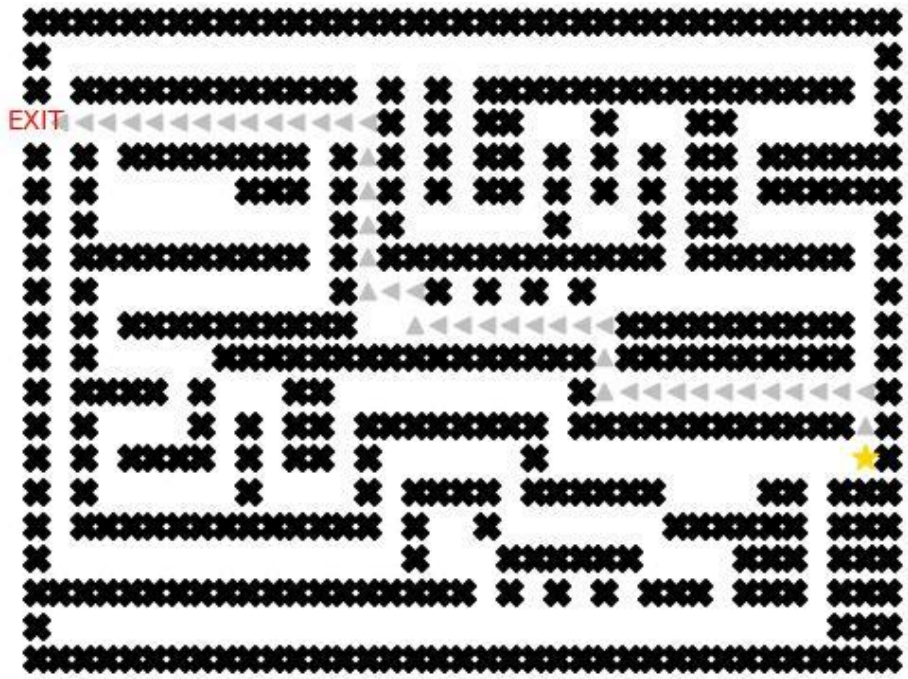
BFS (chi phí 45)



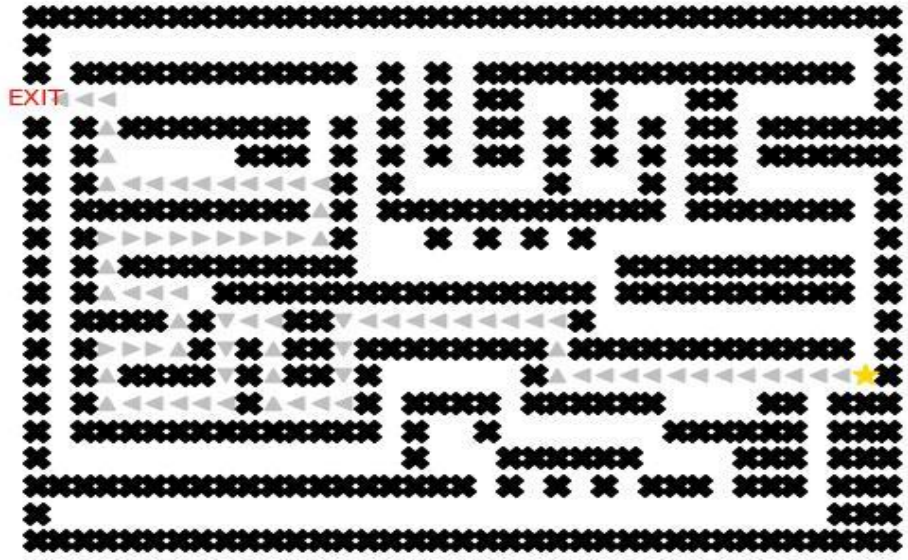
DFS (185)



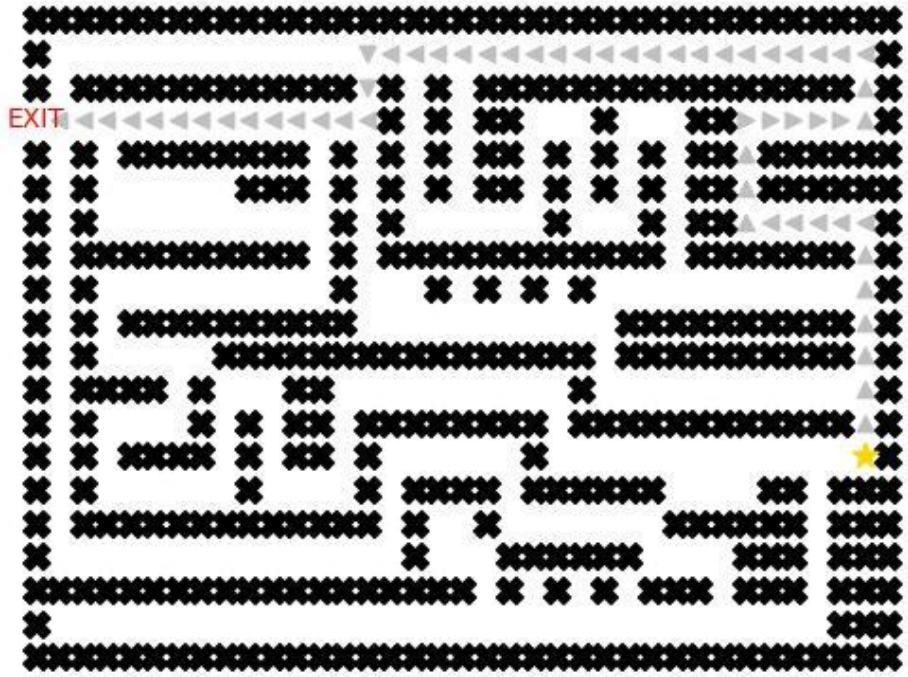
UCS (45)



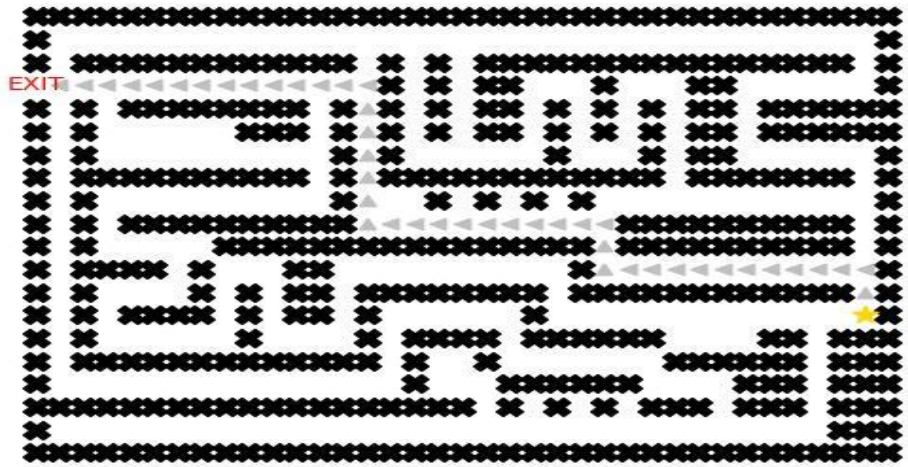
GBFS_H1 (81)



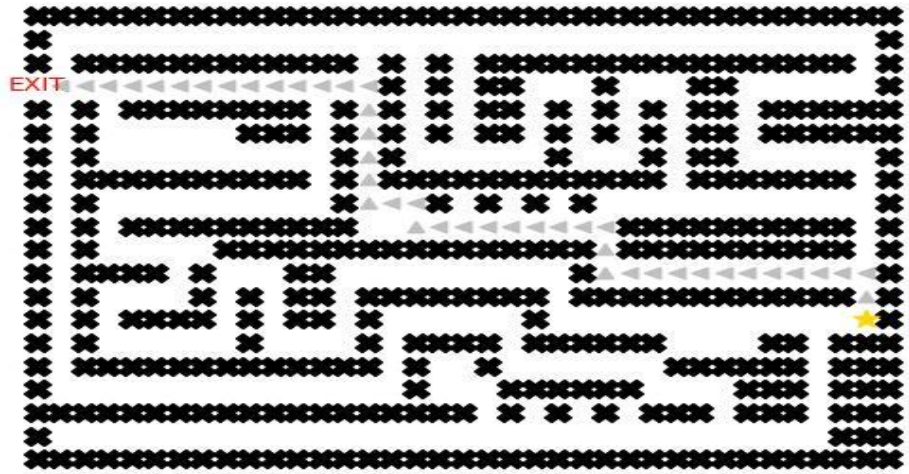
GBFS_H2(59)



ASTAR_H1 (45)



ASTAR_H2 (45)



Ở các bản đồ, UCS và BFS cho kết quả tìm kiếm như nhau (vì chi phí di chuyển đều bằng 1)

Ở bản đồ 1, DFS có đường đi dài hơn và thời gian tìm kiếm lâu hơn BFS, Astar_h1 cho kết quả giống Astar_h2 và giống với BFS và GBFS_h1, GBFS_h2 cho đường đi dài hơn GBFS_h1.

V. Thuật toán đề xuất cho bản đồ có điểm thưởng

Thuật toán sử dụng kết hợp hàm heuristic đánh giá theo khoảng cách Manhattan và thuật toán GBFS để tìm ra đường đi tối ưu.

+ Bước 1: Duyệt qua các điểm thưởng trong danh sách điểm thưởng, xác định khoảng cách ước tính theo heuristic tính từ điểm thưởng đến điểm nguồn và đỉnh đích lần lượt gọi là a và b.

+ Bước 2: Cài đặt hàm đánh giá f(n) = a + b + point (điểm thưởng của điểm đó)

+ Bước 3: Đưa các điểm thưởng vào hàng đợi ưu tiên theo hàm đánh giá f(n)

+ Bước 4: Lấy điểm thưởng đầu tiên ra, nếu điểm thưởng nằm trong vùng chữ nhật tạo bởi điểm nguồn và đích thì sử dụng thuật toán GBFS để tìm đường đi từ đỉnh nguồn đến điểm thưởng p, nếu điểm thưởng nằm ngoài vùng chữ nhật tạo bởi điểm nguồn và đích và bỏ qua và lấy tiếp điểm thưởng đầu tiên trong hàng

đợi ưu tiên và tiếp tục xét. Nếu hàng đợi rỗng, sử dụng thuật toán gbfs tìm kiếm đường đi từ đỉnh hiện tại đến đỉnh đích

+ Bước 5: Nếu đã có đường đi từ đỉnh nguồn đến đỉnh p thì xét tiếp các điểm thưởng còn lại trong hàng đợi ưu tiên, nếu điểm nào có khoảng cách ước tính đến đích cộng với giá trị điểm thưởng cộng với khoảng cách ước tính từ đỉnh hiện tại đến điểm thưởng đang xét thì sử dụng GBFS tìm đường đi đến điểm thưởng đó, nếu hàng đợi ưu tiên chưa rỗng, tiếp tục lặp lại bước 5

+ Bước 6: Hàng đợi ưu tiên đã rỗng, sử dụng GBFS tìm đường đi từ đỉnh hiện tại đến đỉnh đích.

VI. Tham khảo

https://colab.research.google.com/drive/1ejLc4LkrmjpbcRYC3W2xjfA0C0o1PWTp?usp=sharing