

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

=====***=====



BÁO CÁO BTL THUỘC HỌC PHẦN
TRÍ TUỆ NHÂN TẠO

TÌM HIỂU THUẬT TOÁN TÌM KIẾM MÙ VÀ
ỨNG DỤNG VÀO BÀI TOÁN TÌM ĐƯỜNG ĐI
NGẮN NHẤT TRONG MÊ CUNG

GVHD: ThS. Mai Thanh Hồng
Nhóm – Lớp: 8 - 2024IIT6094007
Thành viên: Trần Nguyễn Trà Giang - 2023600740
Hoàng Xuân Hưng - 2023601662
Tạ Tuấn Nguyên - 2023601547
Phùng Bảo Trường - 2023600462

Hà Nội, 2025

BÁO CÁO HỌC TẬP NHÓM

Tên lớp: 20242IT6094003

Khoá: 18

Tên nhóm: 8

Họ và tên thành viên trong nhóm:

- (1)Họ và tên SV: Trần Nguyễn Trà Giang

Mã SV: 2023600740
- (2)Họ và tên SV: Hoàng Xuân Hưng

Mã SV: 2023601662
- (3)Họ và tên SV: Tạ Tuấn Nguyên

Mã SV: 2023601547
- (4)Họ và tên SV: Phùng Bảo Trường

Mã SV: 2023600462

Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung

Tuần	Người thực hiện	Nội dung công việc	Kết quả đạt được	Kiến nghị với giảng viên hướng dẫn (Nêu những khó khăn, hỗ trợ từ phía giảng viên,... nếu cần)
1	Cả nhóm	Viết lời cảm ơn, lời mở đầu, trình bày về không gian trạng thái và thuật toán tìm kiếm mù	Các thành viên hoàn thành đúng thời hạn và đầy đủ nội dung	
2	Cả nhóm	Trình bày về tìm kiếm tối ưu và các thuật toán AKT, AT, A*	Các thành viên hoàn thành đúng thời hạn và đầy đủ nội dung	
3	Cả nhóm	Lựa chọn bài toán, tiến hành mô tả không gian trạng thái và chuyển đổi không gian trạng thái thành đồ thị	Cả nhóm hoàn thành đúng thời hạn	
4	Cả nhóm	Cài đặt chương trình và chạy được chương trình	Cả nhóm đã thống nhất về các biến và xây dựng được thuật toán	
5	Cả nhóm	Hoàn thiện báo cáo và các biểu mẫu, phiếu liên quan	Cả nhóm hoàn thành đầy đủ và đúng thời hạn	

Ngày tháng năm 2025

XÁC NHẬN CỦA GIẢNG VIÊN

(Ký, ghi rõ họ tên)

PHIẾU HỌC TẬP NHÓM

I. Thông tin chung:

1. Tên lớp: 20242IT6094003 Khoá: 18
2. Tên nhóm: 8
3. Họ và tên thành viên trong nhóm:

(5) Họ và tên SV: Trần Nguyễn Trà Giang	Mã SV: 2023600740
(6) Họ và tên SV: Hoàng Xuân Hưng	Mã SV: 2023601662
(7) Họ và tên SV: Tạ Tuấn Nguyên	Mã SV: 2023601547
(8) Họ và tên SV: Phùng Bảo Trường	Mã SV: 2023600462

II. Nội dung học tập:

1. Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung

2. Hoạt động của sinh viên:

- Hoạt động 1: Đề xuất chủ đề nghiên cứu

+ Nội dung:

- Viết đề xuất lựa chọn chủ đề nghiên cứu và xin ý kiến người hướng dẫn về chủ đề nghiên cứu
- Lập biên bản họp và làm việc nhóm (theo BM03)
- Đặt ra các quy tắc làm việc nhóm

+ Mục tiêu/chuẩn đầu ra: Lập biên bản họp và làm việc nhóm

- Hoạt động 2: Báo cáo tiến độ lần 1

Mục tiêu/chuẩn đầu ra:

- Viết được nội dung phần mở đầu, cảm ơn và Chương 1
- Chương 1: Không gian trạng thái và các thuật toán tìm kiếm.
- Giới thiệu tổng quan về không gian trạng thái, toán tử chuyển trạng thái, tìm hiểu về thuật toán tìm kiếm theo chiều sâu (DFS), thuật toán tìm kiếm theo chiều rộng (BFS), thuật toán Heuristic.

- Hoạt động 3: Báo cáo tiến độ lần 2

Mục tiêu/chuẩn đầu ra:

- Viết được nội dung Chương 2: Xây dựng chương trình.

- Tiến hành mô tả bài toán đã được đưa ra và sử dụng ngôn ngữ lập trình Python để ứng dụng thuật toán tìm kiếm mù để tìm đường đi ngắn nhất trong mê cung.
- Hoạt động 4: Nộp cuốn báo cáo thí nghiệm/thực nghiệm và chương trình code

Mục tiêu/chuẩn đầu ra: Hoàn thành và nộp sản phẩm nghiên cứu

3. Sản phẩm nghiên cứu: Quyền báo cáo thí nghiệm/thực nghiệm và chương trình code giải quyết bài toán tìm đường đi ngắn nhất trong mê cung

III. Nhiệm vụ học tập:

1. Hoàn thành Tiểu luận, Bài tập lớn, Đồ án/Dự án theo đúng thời gian quy định (từ ngày 09/04/2025 đến ngày 26/05/2025)
2. Báo cáo sản phẩm nghiên cứu theo chủ đề được giao trước giảng viên và những sinh viên khác

IV. Học liệu thực hiện Bài tập lớn:

1. Tài liệu học tập:

[1] Nguyễn Phương Nga, Trần Hùng Cường, *Giáo trình Trí tuệ nhân tạo*, NXB Thống kê, 2021.

[2] Nguyễn Thanh Thủy, *Trí tuệ nhân tạo*, NXB Thống kê, 1999.

[3] Nguyễn Đình Thức, *Giáo trình Trí tuệ nhân tạo: Mạng nơron phương pháp và ứng dụng*, NXB Giáo dục, 2000.

2. Phương tiện, nguyên liệu thực hiện Bài tập lớn: Máy tính cá nhân, máy chiếu, mạng internet.

KẾ HOẠCH LÀM VIỆC NHÓM								
1. Tên lớp: 20242IT6094003 Khóa: 18								
2. Nhóm: 8								
3. Ngày bắt đầu: 09/04/2025								
4. Ngày kết thúc: //2025								
5. Thành viên nhóm:								
(9) Họ và tên SV: Trần Nguyễn Trà Giang						Mã SV: 2023600740		
(10) Họ và tên SV: Hoàng Xuân Hưng						Mã SV: 2023601662		
(11) Họ và tên SV: Tạ Tuấn Nguyên						Mã SV: 2023601547		
(12) Họ và tên SV: Phùng Bảo Trường						Mã SV: 2023600462		
#	Công việc	Ngày bắt đầu dự kiến	Ngày bắt đầu thực tế	Ngày kết thúc dự kiến	Ngày kết thúc thực tế	Trạng thái	Người thực hiện	Ghi chú
1	Viết lời mở đầu và lời cảm ơn	09/04 /2025	09/04 /2025	13/04 /2025	13/04 /2025	Done	Tạ Tuấn Nguyên	
2	Trình bày về không gian trạng thái	09/04 /2025	09/04 /2025	13/04 /2025	13/04 /2025	Done	Hoàng Xuân Hưng	
3	Trình bày về thuật toán tìm kiếm mù	09/04 /2025	09/04 /2025	13/04 /2025	13/04 /2025	Done	Phùng Bảo Trường, Trần Nguyễn Trà Giang	
4	Trình bày về tìm kiếm tối ưu	14/04 /2025	14/04 /2025	19/04 /2025	19/04 /2025	Done	Hoàng Xuân Hưng	

5	Trình bày về thuật toán AT	14/04/2025	14/04/2025	19/04/2025	19/04/2025	Done	Trần Nguyễn Trà Giang	
6	Trình bày về thuật toán AKT	14/04/2025	14/04/2025	19/04/2025	19/04/2025	Done	Tạ Tuấn Nguyên	
7	Trình bày về thuật toán A*	14/04/2025	14/04/2025	19/04/2025	19/04/2025	Done	Phùng Bảo Trường	
8	Mô tả không gian trạng thái của bài toán	23/04/2025	23/04/2025	26/04/2025	26/04/2025	Done	Trần Nguyễn Trà Giang, Hoàng Xuân Hưng, Tạ Tuấn Nguyên, Phùng Bảo Trường	
9	Chuyển đổi không gian trạng thái thành đồ thị	23/04/2025	23/04/2025	26/04/2025	26/04/2025	Done	Trần Nguyễn Trà Giang, Hoàng Xuân Hưng, Tạ Tuấn Nguyên, Phùng Bảo Trường	
10	Cài đặt chương bằng thuật toán tìm kiếm BFS	28/04/2025	28/04/2025	02/05/2025	02/05/2025	Done	Trần Nguyễn Trà Giang, Hoàng Xuân Hưng, Tạ Tuấn Nguyên, Phùng	

							Bảo Trưởng	
11	Thử nghiệm và chạy demo chương trình	28/04 /2025	28/04 /2025	02/05 /2025	02/05 /2025	Done	Trần Nguyễn Trà Giang, Hoàng Xuân Hưng, Tạ Tuấn Nguyễn, Phùng Bảo Trưởng	
12	Hoàn thiện báo cáo và các biểu mẫu, phiếu liên quan	06/05 /2025	06/05 /2025	16/05 /2025	16/05 /2025	Done	Trần Nguyễn Trà Giang, Hoàng Xuân Hưng, Tạ Tuấn Nguyễn, Phùng Bảo Trưởng	

BIÊN BẢN HỌP, LÀM VIỆC NHÓM TUẦN 1					
1. Tên lớp: 20242IT6094003 Khóa: 18					
2. Nhóm: 8					
3. Thời gian: 09/04/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Trần Nguyễn Trà Giang					
6. Thành viên tham dự - Participants: Trần Nguyễn Trà Giang, Hoàng Xuân Hưng, Tạ Tuấn Nguyên, Phùng Bảo Trường					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Trần Nguyễn Trà Giang	✓			Done
2	Hoàng Xuân Hưng	✓			Done
3	Tạ Tuấn Nguyên	✓			Done
4	Phùng Bảo Trường	✓			Done
Chương trình họp - Meeting agenda					
#	Mục nội dung - Items	Người trình bày - Owner(s)	Thời gian - Time	Ghi chú, trao đổi - Note	
1	Quản lý nhóm	Cả nhóm	15p	- Bầu nhóm trưởng mỗi tuần họp	
2	Thiết lập kênh giao tiếp, lưu trữ	Hoàng Xuân Hưng	5p	- Kênh giao tiếp của cả nhóm thông qua: Messenger - Kênh lưu trữ: Google drive	
3	Xác định mục tiêu làm việc nhóm	Cả nhóm	5p	- Tham gia đầy đủ và đúng giờ các cuộc họp - Vắng có lý do hợp lý - Có trách nhiệm hoàn thành công việc đã được giao	
4	Tìm hiểu và thống nhất đề tài	Cả nhóm	30p	Thống nhất đề tài làm BTL	
5	Xác định các yêu cầu của đề tài	Cả nhóm	10p	Tìm hiểu, nghiên cứu các thuật toán tìm kiếm mù và sử dụng ngôn ngữ Python để ứng dụng vào bài toán tìm đường đi ngắn nhất trong mê cung	
6	Phân chia công việc cho từng thành viên	Trần Nguyễn Trà Giang	10p	- Viết lời mở đầu và lời cảm ơn: Tạ Tuấn Nguyên	

				- Trình bày về không gian trạng thái: Hoàng Xuân Hưng - Trình bày về thuật toán tìm kiếm mù theo chiều sâu: Phùng Bảo Trường - Trình bày về thuật toán tìm kiếm mù theo chiều rộng: Trần Nguyễn Trà Giang	
Vấn đề & Giải pháp - Issues/problems & Solutions					
#	Vấn đề - Issues/problems	Các giải pháp đề xuất - Suggested solutions	Giải pháp được chọn - Selected solution	Ghi chú - Notes	
1	Xác định đề tài	Thuật toán mù, ID3, Naïve bayes	Thuật toán mù	Đề tài ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung	
2	Xác định ngôn ngữ lập trình	Ưu tiên lựa chọn ngôn ngữ lập trình đã học và dễ hiểu	Ưu tiên lựa chọn ngôn ngữ lập trình đã học và dễ hiểu	Ngôn ngữ Python	
Kế hoạch hoạt động - Action plan					
#	Hoạt động - Action	Thời hạn - Deadline	Người thực hiện - Owner(s)	Ghi chú, trao đổi - Notes	
1	Viết lời mở đầu và lời cảm ơn	13/04/2025	Tạ Tuấn Nguyên	Hoàn thành việc trình bày lời mở đầu và cảm ơn	
2	Trình bày về không gian trạng thái	13/04/2025	Hoàng Xuân Hưng	Hoàn thành nội dung về không gian trạng thái	
3	Trình bày về thuật toán tìm kiếm mù theo chiều sâu	13/04/2025	Phùng Bảo Trường	Hoàn thành nội dung về thuật toán tìm kiếm mù theo chiều sâu	
4	Trình bày về thuật toán tìm kiếm mù theo chiều rộng	13/04/2025	Trần Nguyễn Trà Giang	Hoàn thành nội dung về thuật toán tìm kiếm mù theo chiều rộng	
Đóng góp nhóm - Team contribution					
#	Thành viên - Member	Ý tưởng, giải pháp - Ideas	Hỗ trợ người khác	Hoạt động xây dựng nhóm -	Ghi chú - Notes

			- Support other(s)	Team building activities	
1	Trần Nguyễn Trà Giang	3	1	1	
2	Hoàng Xuân Hưng	2	1	1	
3	Tạ Tuấn Nguyên	3	1	1	
4	Phùng Bảo Trường	2	1	1	
Kết quả đánh giá phản hồi của nhóm - Team feedback					
#	Số phiếu 4	Số phiếu 3	Số phiếu 2	Số phiếu 1	
1	4	0	0	0	

BIÊN BẢN HỌP, LÀM VIỆC NHÓM TUẦN 2					
1. Tên lớp: 20242IT6094003 Khóa: 18					
2. Nhóm: 8					
3. Thời gian: 13/04/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Hoàng Xuân Hưng					
6. Thành viên tham dự - Participants: Trần Nguyễn Trà Giang, Hoàng Xuân Hưng, Tạ Tuấn Nguyên, Phùng Bảo Trường					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Trần Nguyễn Trà Giang	✓			Done
2	Hoàng Xuân Hưng	✓			Done
3	Tạ Tuấn Nguyên	✓			Done
4	Phùng Bảo Trường	✓			Done
Chương trình họp - Meeting agenda					
#	Mục nội dung - Items	Người trình bày - Owner(s)	Thời gian - Time	Ghi chú, trao đổi - Note	
1	Check tiến độ	Trần Nguyễn Trà Giang	30p	Tất cả thành viên hoàn thành nhiệm vụ đúng hạn	
2	Phân chia công việc cho từng thành viên	Hoàng Xuân Hưng	30p	-Trình bày về thuật toán tối ưu(Hoàng Xuân Hưng) -Trình bày về thuật toán AT(Trần Nguyễn Trà Giang) -Trình bày về thuật toán AKT(Tạ Tuấn Nguyên) -Trình bày về thuật toán A*(Phùng Bảo Trường)	
Vấn đề & Giải pháp - Issues/problems & Solutions					

#	Vấn đề - Issues/problems	Các giải pháp đề xuất - Suggested solutions	Giải pháp được chọn - Selected solution	Ghi chú - Notes	
1	Chưa tìm được áp dụng thực tế cho bài toán	Tham khảo, tìm hiểu chương trình có sẵn, xem các tài liệu trên học kết hợp	Tham khảo, tìm hiểu chương trình có sẵn, xem các tài liệu trên học kết hợp		
Kế hoạch hoạt động - Action plan					
#	Hoạt động - Action	Thời hạn - Deadline	Người thực hiện - Owner(s)	Ghi chú, trao đổi - Notes	
1	Trình bày về thuật toán tối ưu	19/04/2025	Hoàng Xuân Hưng	Hoàn thành việc trình bày thuật toán tối ưu	
2	Trình bày về thuật toán AKT	19/04/2025	Tạ Tuấn Nguyễn	Hoàn thành trình bày về thuật toán AKT	
3	Trình bày về thuật toán A*	19/04/2025	Phùng Bảo Trương	Hoàn thành nội dung về trình bày về thuật toán A*	
4	Trình bày về thuật toán AT	19/04/2025	Trần Nguyễn Trà Giang	Hoàn thành nội dung về trình bày về thuật toán AT	
Đóng góp nhóm - Team contribution					
#	Thành viên - Member	Ý tưởng, giải pháp - Ideas	Hỗ trợ người khác - Support other(s)	Hoạt động xây dựng nhóm - Team building activities	Ghi chú - Notes
1	Trần Nguyễn Trà Giang	2	0	1	
2	Hoàng Xuân Hưng	1	0	1	
3	Tạ Tuấn Nguyễn	1	0	1	
4	Phùng Bảo Trường	2	0	1	
Kết quả đánh giá phản hồi của nhóm - Team feedback					
#	Số phiếu 4	Số phiếu 3	Số phiếu 2	Số phiếu 1	

1	4	0	0	0	
---	---	---	---	---	--

BIÊN BẢN HỌP, LÀM VIỆC NHÓM TUẦN 3					
1. Tên lớp: 20242IT6094003 Khóa: 18					
2. Nhóm: 8					
3. Thời gian: 23/04/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Tạ Tuấn Nguyên					
6. Thành viên tham dự - Participants: Trần Nguyễn Trà Giang, Hoàng Xuân Hưng, Tạ Tuấn Nguyên, Phùng Bảo Trường					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Trần Nguyễn Trà Giang	✓			Done
2	Hoàng Xuân Hưng	✓			Done
3	Tạ Tuấn Nguyên	✓			Done
4	Phùng Bảo Trường	✓			Done
Chương trình họp - Meeting agenda					
#	Mục nội dung - Items	Người trình bày - Owner(s)	Thời gian - Time	Ghi chú, trao đổi - Note	
1	Check tiến độ báo cáo	Tạ Tuấn Nguyên	30p	Tất cả thành viên hoàn thành nhiệm vụ đúng hạn	
2	Trao đổi chương 2: Xây dựng chương trình phân mô tả bài toán	Cả nhóm	30p		
Vấn đề & Giải pháp - Issues/problems & Solutions					
#	Vấn đề - Issues/problems	Các giải pháp đề xuất - Suggested solutions		Giải pháp được chọn - Selected solution	Ghi chú - Notes

1	Đưa ra bài toán cụ thể để áp dụng thuật toán	Tham khảo các bài toán trên mạng	Chọn bài toán: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung		
Kế hoạch hoạt động - Action plan					
#	Hoạt động - Action	Thời hạn - Deadline	Người thực hiện - Owner(s)	Ghi chú, trao đổi - Notes	
1	Mô tả không gian trạng thái của bài toán	26/04/2025	Cả nhóm		
2	Chuyển đổi không gian trạng thái thành đồ thị	26/04/2025	Cả nhóm		
Đóng góp nhóm - Team contribution					
#	Thành viên - Member	Ý tưởng, giải pháp - Ideas	Hỗ trợ người khác - Support other(s)	Hoạt động xây dựng nhóm - Team building activities	Ghi chú - Notes
1	Trần Nguyễn Trà Giang	2	1	1	
2	Hoàng Xuân Hưng	2	1	1	
3	Tạ Tuấn Nguyên	2	1	1	
4	Phùng Bảo Trường	2	1	1	
Kết quả đánh giá phản hồi của nhóm - Team feedback					
#	Số phiếu 4	Số phiếu 3	Số phiếu 2	Số phiếu 1	
1	4	0	0	0	

BIÊN BẢN HỌP, LÀM VIỆC NHÓM TUẦN 4					
1. Tên lớp: 20242IT6094003 Khóa: 18					
2. Nhóm: 8					
3. Thời gian: 28/04/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Phùng Bảo Trường					
6. Thành viên tham dự - Participants: Trần Nguyễn Trà Giang, Hoàng Xuân Hưng, Tạ Tuấn Nguyên, Phùng Bảo Trường					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Trần Nguyễn Trà Giang	✓			Done
2	Hoàng Xuân Hưng	✓			Done
3	Tạ Tuấn Nguyên	✓			Done
4	Phùng Bảo Trường	✓			Done
Chương trình họp - Meeting agenda					
#	Mục nội dung - Items	Người trình bày - Owner(s)	Thời gian - Time	Ghi chú, trao đổi - Note	
1	Check tiến độ báo cáo	Phùng Bảo Trường	30p	Tất cả thành viên hoàn thành nhiệm vụ đúng hạn	
2	Trao đổi về vấn đề sử dụng ngôn ngữ lập trình nào để áp dụng vào thuật toán với đề tài đã đưa ra	Cả nhóm	30p	Sử dụng ngôn ngữ Python	
Vấn đề & Giải pháp - Issues/problems & Solutions					
#	Vấn đề - Issues/problems	Các giải pháp đề xuất - Suggested solutions		Giải pháp được chọn - Selected solution	Ghi chú - Notes
1	Cài đặt chương trình và chạy được chương trình	Tham khảo các bài toán trên mạng		Chọn bài toán: Ứng dụng thuật toán mù để tìm	

			đường đi ngắn nhất trong mê cung		
Kế hoạch hoạt động - Action plan					
#	Hoạt động - Action	Thời hạn - Deadline	Người thực hiện - Owner(s)	Ghi chú, trao đổi - Notes	
1	Cài đặt chương trình bằng thuật toán tìm kiếm BFS	02/05/2025	Cả nhóm		
2	Thử nghiệm và chạy demo chương trình	02/05/2025	Cả nhóm		
Đóng góp nhóm - Team contribution					
#	Thành viên - Member	Ý tưởng, giải pháp - Ideas	Hỗ trợ người khác - Support other(s)	Hoạt động xây dựng nhóm - Team building activities	Ghi chú - Notes
1	Trần Nguyễn Trà Giang	2	1	0	
2	Hoàng Xuân Hưng	2	1	0	
3	Tạ Tuấn Nguyên	2	1	0	
4	Phùng Bảo Trường	2	1	0	
Kết quả đánh giá phản hồi của nhóm - Team feedback					
#	Số phiếu 4	Số phiếu 3	Số phiếu 2	Số phiếu 1	
1	4	0	0	0	

BIÊN BẢN HỌP, LÀM VIỆC NHÓM TUẦN 5					
1. Tên lớp: 20242IT6094003 Khóa: 18					
2. Nhóm: 8					
3. Thời gian: 06/05/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Trần Nguyễn Trà Giang					
6. Thành viên tham dự - Participants: Trần Nguyễn Trà Giang, Hoàng Xuân Hưng, Tạ Tuấn Nguyên, Phùng Bảo Trường					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Trần Nguyễn Trà Giang	✓			Done
2	Hoàng Xuân Hưng	✓			Done
3	Tạ Tuấn Nguyên	✓			Done
4	Phùng Bảo Trường	✓			Done
Chương trình họp - Meeting agenda					
#	Mục nội dung - Items	Người trình bày - Owner(s)	Thời gian - Time	Ghi chú, trao đổi - Note	
1	Check tiến độ báo cáo	Trần Nguyễn Trà Giang	30p	Tất cả thành viên hoàn thành nhiệm vụ đúng hạn	
2	Hoàn thiện báo cáo và các biểu mẫu, phiếu liên quan	Cả nhóm	40p		
Vấn đề & Giải pháp - Issues/problems & Solutions					
#	Vấn đề - Issues/problems	Các giải pháp đề xuất - Suggested solutions		Giải pháp được chọn - Selected solution	Ghi chú - Notes
Kế hoạch hoạt động - Action plan					

#	Hoạt động - Action	Thời hạn - Deadline	Người thực hiện - Owner(s)	Ghi chú, trao đổi - Notes	
1	Hoàn thiện báo cáo và các biểu mẫu, phiếu liên quan	16/05/2025	Cả nhóm		
Đóng góp nhóm - Team contribution					
#	Thành viên - Member	Ý tưởng, giải pháp - Ideas	Hỗ trợ người khác - Support other(s)	Hoạt động xây dựng nhóm - Team building activities	Ghi chú - Notes
1	Trần Nguyễn Trà Giang	1	0	0	
2	Hoàng Xuân Hưng	1	0	0	
3	Tạ Tuấn Nguyên	1	0	0	
4	Phùng Bảo Trường	1	0	0	
Kết quả đánh giá phản hồi của nhóm - Team feedback					
#	Số phiếu 4	Số phiếu 3	Số phiếu 2	Số phiếu 1	
1	4	0	0	0	

LỜI CẢM ƠN

Trước tiên, chúng em xin gửi lời cảm ơn sâu sắc đến Trường Đại học Công nghiệp Hà Nội vì đã đưa môn học “Trí tuệ nhân tạo” vào chương trình giảng dạy. Với tất cả lòng biết ơn chân thành, chúng em xin bày tỏ sự trân trọng đối với các thầy cô Khoa Công nghệ Thông tin – những người đã tận tụy truyền đạt tri thức và nhiệt huyết, giúp chúng em tiếp thu những kiến thức quý báu trong suốt quá trình học tập tại trường.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành đến cô Mai Thanh Hồng, giảng viên bộ môn “Trí tuệ nhân tạo”, vì đã tận tâm hướng dẫn, chia sẻ những kiến thức giá trị trong suốt thời gian vừa qua. Nhờ sự chỉ dạy tận tình của cô, chúng em không chỉ học được nhiều điều bổ ích mà còn xây dựng được tinh thần học tập nghiêm túc và hiệu quả. Những kiến thức này chắc chắn sẽ trở thành hành trang quý báu giúp chúng em tự tin bước vào tương lai.

Dù đã rất cố gắng trong quá trình thực hiện, nhưng do kiến thức còn hạn chế và thời gian triển khai có phần gấp rút, chắc chắn bài báo cáo của chúng em không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự đóng góp ý kiến quý báu từ quý thầy cô để bài làm được hoàn thiện hơn.

Chúng em xin trân thành cảm ơn!

MỤC LỤC

DANH MỤC HÌNH ẢNH	1
DANH MỤC BẢNG BIỂU	2
LỜI MỞ ĐẦU	3
CHƯƠNG 1: KHÔNG GIAN TRẠNG THÁI VÀ CÁC THUẬT TOÁN TÌM KIẾM.....	4
1.1 Cơ bản về thuật toán về thuật toán tìm kiếm mù.....	4
1.1.1 Khái niệm	4
1.1.2 Nguồn gốc và phát triển	4
1.1.3 Các bước cơ bản của thuật toán tìm kiếm mù.....	4
1.2 Không gian trạng thái	4
1.2.1 Mô tả trạng thái	4
1.2.2 Toán tử chuyển đổi trạng thái.....	5
1.2.3 Không gian trạng thái của bài toán	5
1.3 Các thuật toán tìm kiếm mù.....	6
1.3.1 Tìm kiếm theo chiều sâu (Depth First Search).....	6
1.3.2 Tìm kiếm theo chiều rộng (Breadth First Search).....	8
1.4 Thuật toán Heuristic.....	10
1.4.1 Tìm kiếm tối ưu (Best-First-Search)	10
1.4.2 Thuật toán A^T	13
1.4.3 Thuật toán A^{KT}	15
1.4.4 Thuật toán A^*	16
CHƯƠNG 2: XÂY DỰNG CHƯƠNG TRÌNH	20
2.1 Mô tả bài toán	20
2.1.1 Không gian trạng thái	20
2.1.2 Chuyển đổi không gian trạng thái thành đồ thị	21
2.2 Cài đặt thuật toán	21
2.2.1 Hàm khởi tạo màn hình	21
2.2.2 Lớp vẽ mê cung	21
2.2.3 Lớp green để đánh dấu điểm kết thúc	21
2.2.4 Lớp red để đánh dấu điểm bắt đầu	22

2.2.5 Lớp yellow biểu diễn đường đi cuối cùng	22
2.2.6 Hàm thiết lập mê cung	22
2.2.7 Hàm thiết lập đường đi ngắn nhất khi thuật toán tìm kiếm kết thúc... ..	22
2.2.8 Thuật toán BFS (tìm kiếm chiều sâu)	23
2.2.9 Thiết lập cuối cùng trước khi test.....	23
2.3 Demo chương trình	23
KẾT LUẬN	26
TÀI LIỆU THAM KHẢO.....	27

DANH MỤC HÌNH ẢNH

Hình 1.1: ví dụ thuật toán DFS	7
Hình 1.2: Ví dụ thuật toán BFS.....	9
Hình 1.3: Ví dụ tìm kiếm tối ưu	11
Hình 1.4: Ví dụ thuật toán A^T	14
Hình 1.5: Ví dụ thuật giải A^{KT}	16
Hình 1.6: Ví dụ thuật giải A^*	18
Hình 2.1: Mô tả không gian trạng thái	20
Hình 2.2: Chuyển đổi không gian trạng thái thành đồ thị.....	21

DANH MỤC BẢNG BIỂU

Bảng 1.1: Ví dụ thuật giải DFS	7
Bảng 1.2: Ví dụ thuật giải BFS	10
Bảng 1.3: Ví dụ thuật giải A^T	14
Bảng 1.4: Bảng so sánh 2 thuật toán DFS và BFS.....	19

LỜI MỞ ĐẦU

Trong bối cảnh thời đại công nghệ số, trí tuệ nhân tạo (AI) đang ngày càng khẳng định vai trò quan trọng trong hầu hết các lĩnh vực của đời sống, từ y tế, giáo dục, tài chính cho đến giao thông. AI không chỉ giúp tối ưu hóa hiệu suất làm việc mà còn góp phần thay đổi cách con người tư duy, học tập và tương tác với thế giới xung quanh. Một trong những nội dung cốt lõi của trí tuệ nhân tạo chính là các thuật toán tìm kiếm – công cụ nền tảng để giải quyết nhiều bài toán phức tạp trong thực tiễn.

Xuất phát từ niềm hứng thú với các thuật toán tìm kiếm, đặc biệt là thuật toán tìm kiếm mù, chúng em đã lựa chọn “Bài toán tìm đường đi ngắn nhất trong mê cung” làm đề tài thực nghiệm cho bài tập lớn này. Bài toán mê cung không chỉ mang tính chất trực quan và gần gũi mà còn là một ví dụ điển hình để áp dụng hiệu quả các thuật toán tìm kiếm không có thông tin như BFS, DFS,...

Qua việc triển khai và đánh giá hiệu quả của các thuật toán trên cho “Bài toán tìm đường đi ngắn nhất trong mê cung” chúng em có cơ hội hiểu sâu hơn về bản chất hoạt động, ưu – nhược điểm và phạm vi ứng dụng thực tế của chúng.

Nhóm sinh viên thực hiện!

CHƯƠNG 1: KHÔNG GIAN TRẠNG THÁI VÀ CÁC THUẬT TOÁN TÌM KIẾM

1.1 Cơ bản về thuật toán về thuật toán tìm kiếm mù

1.1.1 Khái niệm

- Thuật toán tìm kiếm mù là các thuật toán không sử dụng bất kỳ thông tin nào về trạng thái đích hoặc khoảng cách đến trạng thái đích. Chỉ dựa trên cấu trúc của không gian trạng thái và phương pháp duyệt để tìm kiếm giải pháp.
- Không gian trạng thái: Bao gồm tất cả các trạng thái có thể có của hệ thống hoặc bài toán, các hành động có thể thực hiện, và các chuyển đổi giữa các trạng thái..

1.1.2 Nguồn gốc và phát triển

- Các thuật toán tìm kiếm mù được phát triển từ các nghiên cứu ban đầu về lý thuyết đồ thị và cấu trúc dữ liệu trong khoa học máy tính cũng như để tìm kiếm các giải pháp cho các bài toán phức tạp trong nhiều lĩnh vực như tối ưu hóa, điều khiển, và lập kế hoạch.
- Các thuật toán tìm kiếm mù đã được mở rộng và cải tiến để trở nên hiệu quả hơn trong các bài toán thực tiễn.

1.1.3 Các bước cơ bản của thuật toán tìm kiếm mù

- Bước 1 - Khởi tạo: Bắt đầu từ trạng thái ban đầu và thêm nó vào danh sách hoặc hàng đợi để duyệt.
- Bước 2 - Duyệt các trạng thái: Lặp lại quá trình lấy trạng thái ra khỏi danh sách hoặc hàng đợi, thăm các trạng thái kề và thêm các trạng thái mới chưa được thăm vào danh sách hoặc hàng đợi.
- Bước 3 - Kiểm tra mục tiêu: Kiểm tra xem trạng thái hiện tại có phải là trạng thái mục tiêu không. Nếu có, thuật toán dừng lại và trả về giải pháp.

1.2 Không gian trạng thái

1.2.1 Mô tả trạng thái

- Giải bài toán trong không gian trạng thái, trước hết phải xác định dạng mô tả trạng thái bài toán sao cho bài toán trở nên đơn giản hơn, phù hợp bản chất vật lý của bài toán (Có thể sử dụng các xâu ký hiệu, vectơ, mảng hai chiều, cây, danh sách,...).
- Mỗi trạng thái chính là mỗi hình trạng của bài toán, các tình trạng ban đầu và tình trạng cuối của bài toán gọi là trạng thái đầu và trạng thái cuối.

Ví dụ: trong bài toán tìm đường đi trong mê cung là chỉ có 1 lối vào và 1 lối ra và ta cần phải tìm đường để có thể thoát khỏi mê cung

- Tại mỗi thời điểm cần xác định vị trí hiện tại của nhân vật và hướng đi tiếp theo của nhân vật.
- Gọi (x, y) là tọa độ hay còn gọi là vị trí của nhân vật:
- + Trạng thái ban đầu là : $(0, 0)$
- + Trạng thái cuối là : (m, n) với m và n là tọa độ của đích đến hay còn gọi là lối ra

1.2.2 Toán tử chuyển đổi trạng thái

- Toán tử chuyển trạng thái thực chất là các phép biến đổi đưa từ trạng thái này sang trạng thái khác. Có hai cách dùng để biểu diễn các toán tử:
- Gọi (x, y) là tọa độ hay còn gọi là vị trí của nhân vật:
- + Biểu diễn như một hàm xác định trên tập các trạng thái và nhận giá trị cũng trong tập này.
- + Biểu diễn dưới dạng các quy tắc sản xuất $S \rightarrow A$ có nghĩa là nếu có trạng thái S thì có thể đưa đến trạng thái A .

Ví dụ : Trong bài toán tìm đường trong mê cung :

- Các trạng thái có thể xảy ra : Giả sử 1 lần dịch chuyển sẽ dịch chuyển 1 đơn vị (1 ô).
- + Đi sang trái
- + Đi sang phải
- + Đi xuống
- Như vậy, nếu xét trạng thái tại thời điểm hiện tại là (x, y) thì các trạng thái kế tiếp có thể là :
- + $(x, y - 1)$: đi sang trái
- + $(x, y + 1)$: đi sang phải
- + $(x + 1, y)$: đi xuống

1.2.3 Không gian trạng thái của bài toán

- Không gian trạng thái là tập tất cả các trạng thái có thể có và tập các toán tử của bài toán.
- Không gian trạng thái là một bộ bốn, Ký hiệu: $K = (T, S, G, F)$. Trong đó:
- + T : tập tất cả các trạng thái có thể có của bài toán.
- + S : trạng thái đầu.
- + G : tập các trạng thái đích.

+ F: tập các toán tử.

Trong trường hợp này không gian trạng thái bài toán tìm đường đi trong mê cung sẽ được xác định như sau :

$T = \{ (x, y) , x \text{ khác } m , y \text{ khác } n \}$

$S = (0, 0)$: trạng thái ban đầu và phụ thuộc vào lối vào I

$G = (m, n)$: trạng thái đích vì chỉ có 1 lối ra nên G chỉ có 1 tập con là (m, n)

F = Tập hợp các thao tác di chuyển sang trái, sang phải, đi xuống để có thể thay đổi vị trí của nhân vật

- Ví dụ:

Ta có mê cung: I + 0 0

0 0 0 +

+ + 0 0

0 0 + O

với 0 là đường có thể đi và + là vật cản

Các trạng thái:

- Trạng thái ban đầu: $(0, 0)$
- Trạng thái mục tiêu: $(3, 3)$
- Các trạng thái kề của $(0, 0)$: $(1, 0)$

Thuật toán tìm kiếm mù (ví dụ BFS):

1. Bắt đầu từ trạng thái ban đầu $(0, 0)$.
2. Thêm $(0, 0)$ vào hàng đợi và đánh dấu là đã thăm.
3. Lấy $(0, 0)$ ra khỏi hàng đợi và thêm các trạng thái kề hợp lệ vào hàng đợi: $(1, 0)$.
4. Lặp lại cho đến khi tìm thấy trạng thái mục tiêu $(3, 3)$ hoặc hàng đợi rỗng.

1.3 Các thuật toán tìm kiếm mù

1.3.1 Tìm kiếm theo chiều sâu (Depth First Search)

a. Tư tưởng của chiến lược tìm kiếm theo chiều sâu

- Từ đỉnh xuất phát duyệt một đỉnh kề.
- + Các đỉnh của đồ thị được duyệt theo các nhánh đến nút lá.
- + Nếu chưa tìm thấy đỉnh T_G thì quay lui tới một đỉnh nào đó để sang nhánh khác.
- + Việc tìm kiếm kết thúc khi tìm thấy đỉnh T_G hoặc đã hết các đỉnh.

b. Thuật toán tìm kiếm theo chiều sâu

- Lưu trữ: Sử dụng hai danh sách DONG và MO trong đó:

- + DONG: Chứa các đỉnh đã xét, hoạt động theo kiểu FIFO (*hàng đợi*).
- + MO: chứa các đỉnh đang xét, hoạt động theo kiểu LIFO (*ngăn xếp*).

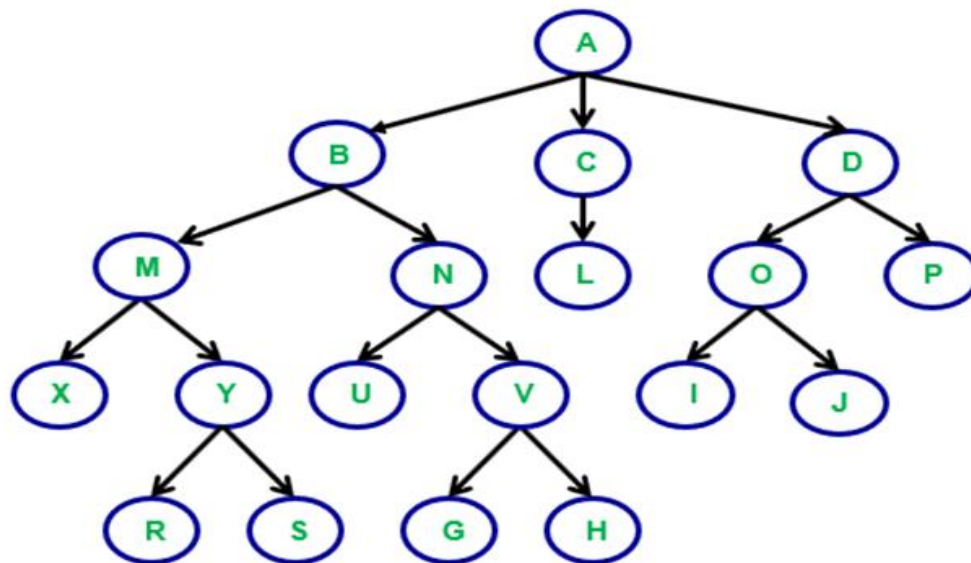
```

1. MO =  $\emptyset$ ;    MO = MO  $\cup$  {T0}
2. while (MO !=  $\emptyset$ )
    {
        n = get(MO)    // lấy đỉnh đầu trong danh sách MO
        if (n==TG)    // nếu n là trạng thái kết thúc
            return TRUE    // tìm kiếm thành công, dừng
        DONG = DONG  $\cup$  {n}    //đánh dấu n đã được xét
        for các đỉnh kề v của n
            if(v chưa đc xét)    //v chưa ở trong DONG
                MO = MO  $\cup$  {v} //đưa v vào đầu danh sách MO
                father(v)=n    // lưu lại vết đường đi từ n đến v
    }

```

c. Ví dụ thuật toán tìm kiếm theo chiều sâu

Cho đồ thị như hình vẽ sau:



Hình 1.1: ví dụ thuật toán DFS

Đỉnh đầu $T_0=A$, $T_G = \{R\}$

Tìm đường đi p từ T_0 đến T_G bằng phương pháp tìm kiếm theo chiều sâu?

Bảng 1.1: Ví dụ thuật giải DFS

n	B(n)	MO	DONG
---	------	----	------

		A	
A	B, C, D	B, C, D	A
B	M, N	M, N, C, D	A,B
M	X, Y	X, Y, N, C, D	A,B,M
X	Ø	Y, N, C, D	A,B,M,X
Y	R, S	R, S, N, C, D	A,B,M,X,Y
R	là đích -> dừng		

Xây dựng đường đi có hành trình: $p = A \rightarrow B \rightarrow M \rightarrow Y \rightarrow R$

Nhận xét:

- + Nếu trong đồ thị G tồn tại đường đi từ T_0 đến 1 đỉnh $T_G \in \text{Goal}$ thì hàm DFS sẽ dừng lại và cho đường đi p có độ dài có thể không ngắn nhất
- + Với DFS các đỉnh được duyệt theo từng nhánh (theo chiều sâu)
- + Thuật toán DFS có độ phức tạp $O(b^d)$ với b là bậc của cây và d là chiều sâu của cây. Tuy nhiên trong trường hợp xấu nhất cũng là $O(b^d)$

d. Ưu điểm và nhược điểm của thuật toán tìm kiếm theo chiều sâu

- Ưu điểm:
 - + Tiết kiệm bộ nhớ: Do chỉ cần lưu trữ trạng thái của 1 nhánh tại 1 thời điểm
 - + Hiệu quả trong không gian trạng thái lớn: DFS có thể nhanh hơn trong việc tìm kiếm giải pháp trong các không gian trạng thái rộng lớn và sâu
- Nhược điểm:
 - + Không đảm bảo tìm được đường đi ngắn nhất: DFS có thể tìm thấy giải pháp nhưng không đảm bảo đó là đường đi ngắn nhất
 - + Dễ bị lặp vô hạn trong không gian trạng thái vô hạn hoặc không có giới hạn độ sâu: cần cơ chế phòng ngừa để tránh lặp vô hạn

1.3.2 Tìm kiếm theo chiều rộng (Breadth First Search)

a. Tư tưởng của chiến lược tìm kiếm theo chiều rộng

- Từ đỉnh xuất phát
- + Thăm tất cả các nút con của nó trước khi chuyển sang thăm các nút con của các nút con đó
- + Ưu tiên thăm tất cả các nút gần gốc hơn trước khi thăm các nút xa hơn.
- + Đánh dấu các nút đã được thăm, đảm bảo không duyệt lại các nút đã thăm, tránh vòng lặp vô tận.

+ Việc tìm kiếm kết thúc khi tìm thấy đỉnh T_G hoặc đã hết các đỉnh.

b. Thuật toán tìm kiếm theo chiều rộng

- Lưu trữ: Sử dụng hai danh sách DONG và MO trong đó:
- + DONG: Chứa các đỉnh đã xét, hoạt động theo kiểu FIFO (*hàng đợi*).
- + MO: chứa các đỉnh đang xét, hoạt động theo kiểu LIFO (*ngăn xếp*).

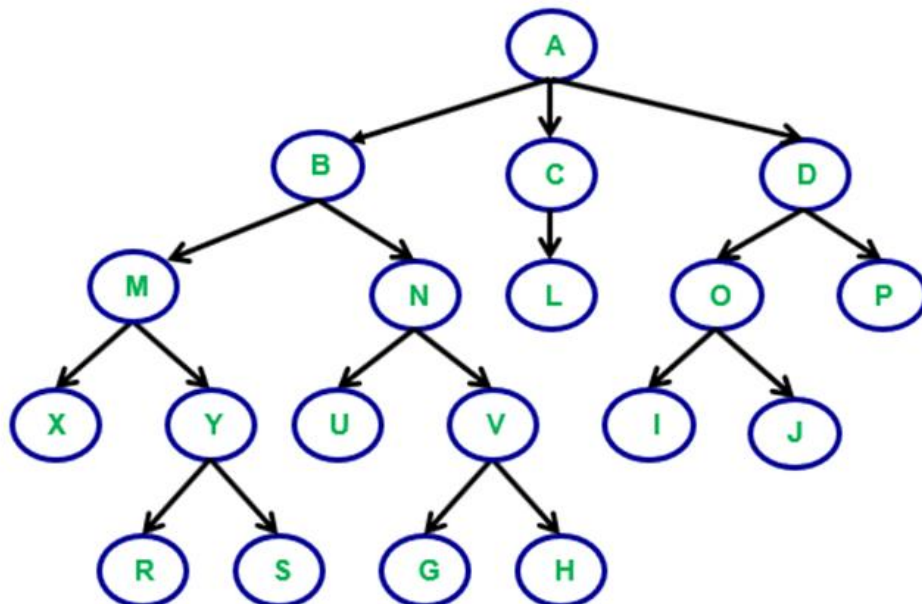
```

1. MO =  $\emptyset$ ;      MO = MO  $\cup$  {T0}
2. while (MO !=  $\emptyset$ )
{
  n = get(MO)    // lấy đỉnh đầu trong danh sách MO
  if (n==TG)      // nếu n là trạng thái kết thúc
    return TRUE    // tìm kiếm thành công, dừng
  DONG = DONG  $\cup$  {n} //đánh dấu n đã được xét
  for các đỉnh kề v của n
    if(v chưa đc xét) //v chưa ở trong DONG
      MO = MO  $\cup$  {v} //đưa v vào cuối danh sách MO
      father(v)=n    // lưu lại vết đường đi từ n đến v
}

```

c. Ví dụ thuật toán tìm kiếm theo chiều rộng

Cho đồ thị như hình vẽ sau:



Hình 1.2: Ví dụ thuật toán BFS

Đỉnh đầu $T_0=A$, $T_G= \{N\}$.

Tìm đường đi p từ T_0 đến T_G bằng phương pháp tìm kiếm theo chiều rộng?

Bảng 1.2: Ví dụ thuật giải BFS

n	B(n)	MO	DONG
		A	
A	B, C, D	B, C, D	A
B	M, N	C, D, M, N	A,B
C	L	D, M, N, L	A,B,C
D	O, P	M, N, L, O, P	A,B,C,D
M	X, Y	N, L, O, P, X, Y	A,B,C,D,M
N	-> là đích -> dừng		

Xây dựng đường đi có hành trình: $p = A \rightarrow B \rightarrow N$.

Nhận xét:

- + Nếu trong đồ thị tồn tại đường đi từ T_0 đến 1 đỉnh $T_G \in \text{Goal}$ thì hàm BFS sẽ dừng lại và cho đường đi p có độ dài ngắn nhất
- + Với BFS các đỉnh được duyệt theo từng mức (theo chiều rộng)
- + Thuật toán BFS có độ phức tạp $O(b^d)$ với b là bậc của cây và d là chiều sâu của cây

c. Ưu điểm và nhược điểm của thuật toán tìm kiếm theo chiều rộng

- Ưu điểm:
 - + Đảm bảo tìm được đường đi ngắn nhất trong đồ thị không có trọng số nếu tồn tại
 - + Đơn giản và dễ hiểu, dễ triển khai
- Nhược điểm:
 - + Tốn kém bộ nhớ: Phải lưu trữ tất cả các đỉnh ở mức hiện tại trước khi chuyển sang mức tiếp theo, dẫn đến việc sử dụng bộ nhớ nhiều
 - + Không hiệu quả trong không gian trạng thái lớn: với số lượng trạng thái rất lớn, BFS có thể trở nên chậm và tốn kém tài nguyên

1.4 Thuật toán Heuristic

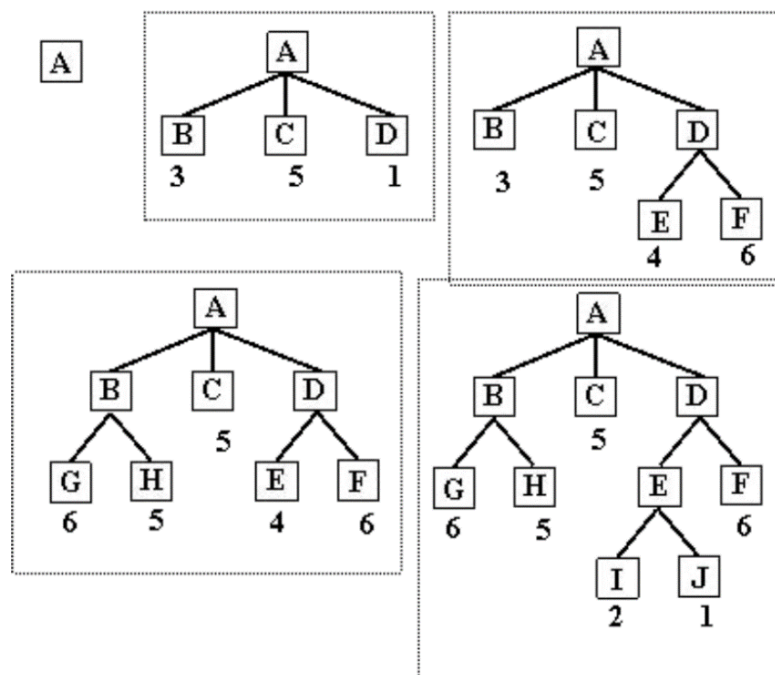
1.4.1 Tìm kiếm tối ưu (Best-First-Search)

Ưu điểm của tìm kiếm theo chiều sâu là không phải quan tâm đến sự mở rộng của tất cả các nhánh. Ưu điểm của tìm kiếm chiều rộng là không bị sa vào các đường dẫn bế tắc (các nhánh cụt). Tìm kiếm tối ưu (Best-First Search-BFS) sẽ kết hợp hai phương pháp trên cho phép ta đi theo một con đường duy nhất tại

một thời điểm, nhưng đồng thời vẫn xét được những hướng khác. Nếu con đường đang đi không triển vọng bằng những con đường đang quan sát, ta sẽ chuyển sang đi theo một trong số các con đường này.

Một cách cụ thể, tại mỗi bước của tìm kiếm BFS, ta chọn đi theo trạng thái có khả năng cao nhất trong số các trạng thái đã được xét cho đến thời điểm đó. BFS khác với tìm kiếm leo đồi là chỉ chọn trạng thái có khả năng cao nhất trong số các trạng thái kế tiếp có thể đến được t trạng thái hiện tại. Như vậy, với tiếp cận này, ta sẽ ưu tiên đi vào những nhánh tìm kiếm có khả năng nhất (giống tìm kiếm leo đồi), nhưng ta sẽ không bị lẫn lộn trong các nhánh này vì nếu càng đi sâu vào một hướng mà ta phát hiện ra rằng hướng này càng đi thì càng xấu, đến mức nó xấu hơn cả những hướng mà ta chưa đi, thì ta sẽ không đi tiếp hướng hiện tại nữa mà chọn đi theo một hướng tốt nhất trong số những hướng chưa đi. Đó là tư tưởng chủ đạo của tìm kiếm tối ưu.

Ví dụ minh họa:



Hình 1.3: Ví dụ tìm kiếm tối ưu

Khởi đầu, chỉ có một nút (trạng thái) A nên nó sẽ được mở rộng tạo ra 3 nút mới B, C và D. Các con số dưới nút là giá trị cho biết độ tốt của nút. Con số càng nhỏ, nút càng tốt. Do D là nút có khả năng nhất nên nó sẽ được mở rộng tiếp sau nút A và sinh ra 2 nút kế tiếp là E và F. Đến đây, ta lại thấy nút B có vẻ có khả năng nhất (trong các nút B, C, E, F) nên ta sẽ chọn mở rộng nút B và tạo ra 2 nút G và H. Nhưng lại một lần nữa, hai nút G, H này được đánh giá ít khả năng hơn

E, vì thế sự chú ý lại trở về E. E được mở rộng và các nút được sinh ra từ E là I và J. Ở bước kế tiếp, J sẽ được mở rộng vì nó có khả năng nhất. Quá trình này tiếp tục cho đến khi tìm thấy một lời giải.

Để cài đặt các thuật giải theo kiểu tìm kiếm BFS, thường cần dùng 2 tập hợp:

- OPEN: tập chứa các trạng thái đã được sinh ra nhưng chưa được xét đến (vì ta đã chọn một trạng thái khác). Thực ra, OPEN là một loại hàng đợi ưu tiên (priority queue) mà trong đó, phần tử có độ ưu tiên cao nhất là phần tử tốt nhất. Người ta thường cài đặt hàng đợi ưu tiên bằng Heap.
- CLOSE: tập chứa các trạng thái đã được xét đến. Chúng ta cần lưu trữ những trạng thái này trong bộ nhớ để đề phòng trường hợp khi một trạng thái mới được tạo ra lại trùng với một trạng thái mà ta đã xét đến trước đó. Trong trường hợp không gian tìm kiếm có dạng cây thì không cần dùng tập này.

Thuật giải

- Đặt OPEN chứa trạng thái khởi đầu.
- Cho đến khi tìm được trạng thái đích hoặc không còn nút nào trong OPEN, thực hiện:
 - + Chọn trạng thái tốt nhất (T_{max}) trong OPEN (và xóa T_{max} khỏi OPEN)
 - + Nếu T_{max} là trạng thái kết thúc thì thoát
 - + Ngược lại, tạo ra các trạng thái kế tiếp T_k có thể có từ trạng thái T_{max} .

Đối với mỗi trạng thái kế tiếp T_k thực hiện:

Tính $f(T_k)$;

Thêm T_k vào OPEN

BFS khá đơn giản. Tuy vậy, trên thực tế, cũng như tìm kiếm chiều sâu và chiều rộng, hiếm khi ta dùng BFS một cách trực tiếp. Thông thường, người ta thường dùng các phiên bản của BFS là A^T , A^{KT} và A^* .

Thông tin về quá khứ và tương lai:

Thông thường, trong các phương án tìm kiếm theo kiểu BFS, chi phí f của một trạng thái được tính dựa theo hai giá trị mà ta gọi là g và h . Trong đó h , như đã biết, đó là một ước lượng về chi phí từ trạng thái hiện hành cho đến trạng thái đích (thông tin tương lai), còn g là chiều dài quãng đường đã đi từ trạng thái ban đầu cho đến trạng thái hiện tại (thông tin quá khứ). Khi đó hàm ước lượng tổng chi phí $f(n)$ được tính theo công thức:

$$f(n) = g(n) + h(n)$$

1.4.2 Thuật toán A^T

Khái niệm

Thuật giải A^T là một phương pháp tìm kiếm theo kiểu BFS với chi phí của đỉnh là giá trị hàm g (tổng chiều dài thực sự của đường đi từ đỉnh bắt đầu đến đỉnh hiện tại).

Cho đồ thị $G = (V, E)$ với V : tập đỉnh; E : Tập cung. Với mỗi một cung người ta gán thêm một đại lượng được gọi là giá của cung.

$$C: E \rightarrow R_+$$

$$e \mapsto C(e)$$

Khi đó đường đi $p = n_1, n_2, \dots, n_k$ có giá được tính theo công thức:

$$C(p) = \sum_{i=1}^{k-1} C(n_i, n_{i+1})$$

Vấn đề đặt ra là tìm đường đi p từ T_0 đến đỉnh $T_G \in \text{Goal}$ sao cho: $C(p)$ là nhỏ nhất.

Thuật toán

Vào: Đồ thị $G = (V, E)$

$$C: E \rightarrow R_+$$

$$e \mapsto C(e)$$

Đỉnh đầu T_0 và Goal chứa tập các đỉnh đích

Ra: Đường đi $p: T_0 \rightarrow T_G \in \text{Goal}$ sao cho:

$$C(p) = g(n_k) = \min \{g(n) / n \in \text{Goal}\}.$$

Phương pháp: Sử dụng hai danh sách DONG và MO

void AT(){

MO = { T_0 }; $g(T_0) = 0$; DONG = \emptyset ;

while (MO != \emptyset) {

n = getNew(MO) // Lấy đỉnh n sao cho $g(n) \rightarrow \min$

if (n == T_G) return True;

else {

for (m \in B(n))

if (m \notin MO) && (m \notin DONG) {

g(m) = $g(n) + \text{cost}(m, n)$; MO = MO \cup {m};

}

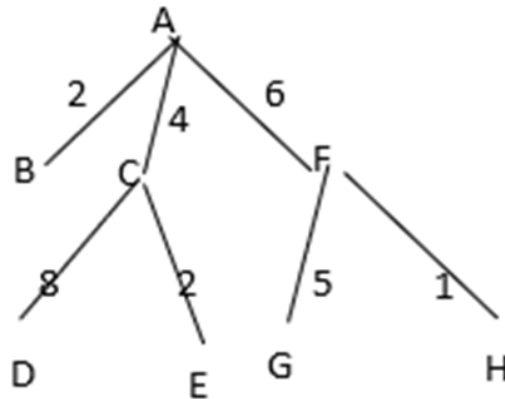
else g(m) = $\min \{g(m), g_{\text{new}}(m)\}$;

```

        DONG = DONGU {n};
    }
}
return False;
}

```

Ví dụ:



Hình 1.4: Ví dụ thuật toán A^T

Bảng 1.3: Ví dụ thuật giải A^T

n	A(n)	OPEN	CLOSE
		A(0)	
A	B, C, F	B(2), C(4), F(6)	A
B	∅	C(4), F(6)	A, B
C	D, E	D(12), E(6), F(6)	A, B, C
E	∅	D(12), F(6)	A, B, C, E
F	G, H	G(11), H(7), D(12)	A, B, C, E, F
H	→ là đích → dừng		

Trong bảng trên các con số trong ngoặc đơn là giá trị của hàm $g()$. Do $H \in \text{Goal}$ nên thuật toán dừng và đường đi tìm được $p: A \rightarrow F \rightarrow H$ có chi phí $C(p) = 7$.

Kết quả: Nếu trong đồ thị G tồn tại đường đi $p: T_0 \rightarrow T_G \in \text{Goal}$ thì thủ tục A^T sẽ dừng và cho kết quả đường đi có độ dài ngắn nhất.

Nhận xét:

+ Nếu $C(a) = 1 \forall a \in E$ thì A^T trở thành BFS

- + Nếu thay điều kiện $g(n) \rightarrow \min$ bằng điều kiện $d(n) \rightarrow \max$ trong đó $d(n)$ là độ sâu hiện tại của đỉnh n . Khi đó A^T trở thành DFS

1.4.3 Thuật toán A^{KT}

Khái niệm

Thuật giải A^T trong quá trình tìm đường đi chỉ xét đến các đỉnh và giá của chúng. Nghĩa là việc tìm đỉnh triển vọng chỉ phụ thuộc hàm $g(n)$ (thông tin quá khứ). Tuy nhiên thuật giải này không còn phù hợp khi gặp phải những bài toán phức tạp (độ phức tạp cấp hàm mũ) do ta phải tháo một lượng nút lớn. Để khắc phục nhược điểm này, người ta sử dụng thêm các thông tin bổ sung xuất phát từ bản thân bài toán để tìm ra các đỉnh có triển vọng, tức là đường đi tối ưu sẽ tập trung xung quanh đường đi tốt nhất nếu sử dụng các thông tin đặc tả về bài toán (thông tin quá tương lai).

Theo thuật giải này, chi phí của đỉnh được xác định:

$$f(n) = g(n) + h(n)$$

Đỉnh n được chọn nếu $f(n)$ đạt min.

Việc xác định hàm ước lượng $h(n)$ được thực hiện dựa theo:

- + Chọn toán tử xây dựng cung sao cho có thể loại bớt các đỉnh không liên quan và tìm ra các đỉnh có triển vọng.
- + Sử dụng thêm các thông tin bổ xung nhằm xây dựng tập OPEN và cách lấy các đỉnh trong tập OPEN.

Để làm được việc này người ta phải đưa ra độ đo, tiêu chuẩn để tìm ra các đỉnh có triển vọng. Các hàm sử dụng các kỹ thuật này gọi là hàm đánh giá. Sau đây là một số phương pháp xây dựng hàm đánh giá:

- + Dựa vào xác suất của đỉnh trên đường đi tối ưu.
- + Dựa vào khoảng cách, sự sai khác của trạng thái đang xét với trạng thái đích hoặc các thông tin liên quan đến trạng thái đích.

Thuật toán

Vào: Đồ thị $G = (V, E)$ trong đó V là tập đỉnh, E là tập cung.

$f: V \rightarrow \mathbb{R}^+$ ($f(n)$: hàm chi phí)

Đỉnh đầu T_0 và tập các đỉnh đích

Ra: Đường đi $p: T_0 \rightarrow T_G \in \text{Goal}$

Phương pháp: Sử dụng 2 danh sách DONG và MO

void $A^{KT}()$ {

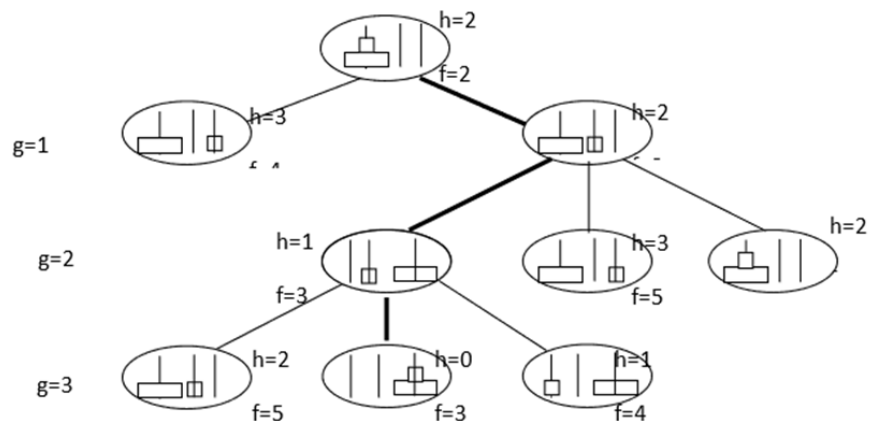
MO = $\{T_0\}$, $g(T_0) = 0$;

```

Tính  $h(T_0)$ ,  $f(T_0) = g(T_0) + h(T_0)$ 
while ( $MO \neq \emptyset$ ) {
     $n = \text{getNew}(MO)$  // lấy đỉnh  $n$  sao cho  $f(n)$  đạt min
    if ( $n == T_G$ ) return True;
    else {
        for ( $m \in B(n)$ ) {
             $g(m) = g(n) + \text{cost}(m,n)$ ;
            Tính  $h(m)$ ,  $f(m) = g(m) + h(m)$ ;
             $OM = MO \cup \{m\}$ ;
        }
    }
}
return False;

```

Ví dụ:



Hình 1.5: Ví dụ thuật giải A^{KT}

1.4.4 Thuật toán A^*

Khái niệm

A^* là một phiên bản đặc biệt của A^{KT} áp dụng cho trường hợp đồ thị. Thuật toán này tìm một đường đi từ một nút khởi đầu tới một nút đích cho trước (hoặc tới một nút thỏa mãn một điều kiện đích). Thuật toán này sử dụng một "đánh giá heuristic" để xếp loại từng nút theo ước lượng về tuyến đường tốt nhất đi qua nút đó. Thuật toán này duyệt các nút theo thứ tự của đánh giá heuristic này.

Ý tưởng trực quan: Xét bài toán tìm đường - bài toán mà A* thường được dùng để giải. A* xây dựng tăng dần tất cả các tuyến đường từ điểm xuất phát cho tới khi nó tìm thấy một đường đi chạm tới đích. Tuy nhiên, cũng như tất cả các thuật toán tìm kiếm có thông tin (*informed tìm kiếm thuật toán*), nó chỉ xây dựng các tuyến đường "có vẻ" dẫn về phía đích.

Để biết những tuyến đường nào có khả năng sẽ dẫn tới đích, A* sử dụng một "đánh giá heuristic" về khoảng cách từ điểm bất kỳ cho trước tới đích. Trong trường hợp tìm đường đi, đánh giá này có thể là khoảng cách đường chim bay - một đánh giá xấp xỉ thường dùng cho khoảng cách của đường giao thông.

Điểm khác biệt của A* đối với tìm kiếm theo lựa chọn tốt nhất là nó còn tính đến khoảng cách đã đi qua. Điều đó làm cho A* "đầy đủ" và "tối ưu", nghĩa là, A* sẽ luôn luôn tìm thấy đường đi ngắn nhất nếu tồn tại một đường đi như vậy. A* không đảm bảo sẽ chạy nhanh hơn các thuật toán tìm kiếm đơn giản hơn. Trong một môi trường dạng mê cung, cách duy nhất để đến đích có thể là trước hết phải đi về phía xa đích và cuối cùng mới quay lại. Trong trường hợp đó, việc thử các nút theo thứ tự "gần đích hơn thì được thử trước" có thể gây tốn thời gian.

Thuật toán A*: A* lưu giữ một tập các lời giải chưa hoàn chỉnh, nghĩa là các đường đi qua đồ thị, bắt đầu từ nút xuất phát. Tập lời giải này được lưu trong một hàng đợi ưu tiên (*priority queue*). Thứ tự ưu tiên gán cho một đường đi được quyết định bởi hàm.

Trong đó, là chi phí của đường đi cho đến thời điểm hiện tại, nghĩa là tổng trọng số của các cạnh đã đi qua. là hàm đánh giá heuristic về chi phí nhỏ nhất để đến đích từ. Ví dụ, nếu "chi phí" được tính là khoảng cách đã đi qua, khoảng cách đường chim bay giữa hai điểm trên một bản đồ là một đánh giá heuristic cho khoảng cách còn phải đi tiếp.

Thuật toán

```
void Asao(){
    MO = {T0}, DONG = ∅, g(T0) = 0, Tinh h(T0), f(T0) = g(T0) + h(T0);
    while(MO != ∅){
        n = getnew(MO) // lấy đỉnh n sao cho f(n) đạt min.
        if (n == TG) return True;
        else {for (m ∈ B(n))
            if (m ∈ (MO ∪ DONG)) {
                Tinh h(m), g(m), f(m) = g(m) + h(m);
```

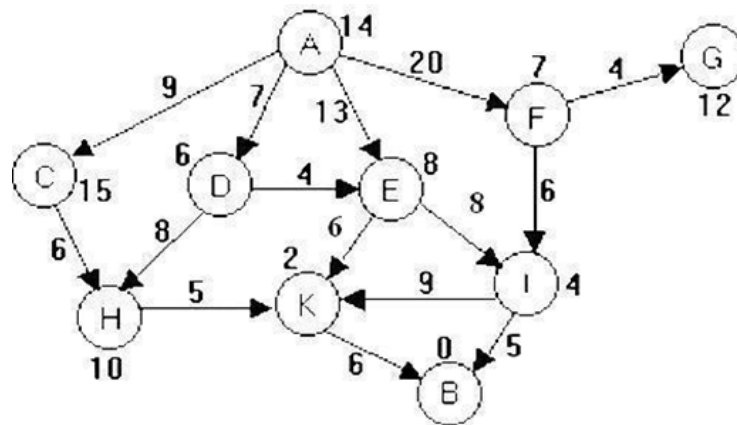
```

        Cha(m) = n; MO = MO  $\cup$  {m};
    }else{g(m) = min {gold(m), gnew(m)};
        Cập nhật lại MO;
    }
}
DONG = DONG  $\cup$  {n};
}
return false;
}

```

Ví dụ:

Trạng thái ban đầu là trạng thái A, trạng thái đích là B, các số ghi cạnh các cung là độ dài đường đi, các số cạnh các đỉnh là giá trị của hàm h



Hình 1.6: Ví dụ thuật giải A^*

Đầu tiên, phát triển đỉnh A sinh ra các đỉnh con C, D, E và F. Tính giá trị của hàm f tại các đỉnh này ta có:

$$g(C) = 9, \quad f(C) = 9 + 15 = 24, \quad g(D) = 7, \quad f(D) = 7 + 6 = 13$$

$$g(E) = 13, \quad f(E) = 13 + 8 = 21, \quad g(F) = 20, \quad f(F) = 20 + 7 = 27$$

Như vậy đỉnh tốt nhất là D (vì $f(D) = 13$ là nhỏ nhất). Phát triển D, ta nhận được các đỉnh con H và E. Ta đánh giá H và E (mới):

$$g(H) = g(D) + \text{Độ dài cung } (D, H) = 7 + 8 = 15, \quad f(H) = 15 + 10 = 25.$$

Đường đi tới E qua D có độ dài:

$$g(E) = g(D) + \text{Độ dài cung } (D, E) = 7 + 4 = 11.$$

$$\text{Vậy đỉnh E mới có đánh giá là } f(E) = g(E) + h(E) = 11 + 8 = 19.$$

Trong số các đỉnh cho phát triển, thì đỉnh E với đánh giá $f(E) = 19$ là đỉnh tốt nhất. Phát triển đỉnh này, ta nhận được các đỉnh con của nó là K và I.

Chúng ta tiếp tục quá trình trên cho tới khi đỉnh được chọn để phát triển là đỉnh kết thúc B, độ dài đường đi ngắn nhất tới B là $g(B) = 19$. Quá trình tìm kiếm trên được mô tả bởi cây tìm kiếm sau, trong đó các số cạnh các đỉnh là các giá trị của hàm đánh giá f .

$$KL: A \rightarrow D \rightarrow E \rightarrow I \rightarrow B$$

Đường đi tìm được có thể không phải là tốt nhất. Nếu $h(n) = 0$ trong mọi trường hợp thì A^* trở thành A^T .

Bảng 1.4: Bảng so sánh 2 thuật toán DFS và BFS

	BFS	DFS
Thứ tự các đỉnh khi duyệt đồ thị	Các đỉnh được duyệt theo từng mức	Các đỉnh được duyệt theo từng nhánh
Độ dài đường đi p từ T_0 đến T_G	Ngắn nhất	Có thể không ngắn nhất
Tính hiệu quả	<ul style="list-style-type: none"> - Chiến lược có hiệu quả khi lời giải nằm ở mức thấp (gần gốc cây) - Thuận lợi khi tìm kiếm nhiều lời giải 	<ul style="list-style-type: none"> - Chiến lược có hiệu quả khi lời giải nằm gần hướng đi được chọn theo phương án - Thuận lợi khi tìm kiếm 1 lời giải
Sử dụng bộ nhớ	Lưu trữ toàn bộ không gian trạng thái	Lưu trữ các trạng thái đang xét
Trường hợp tốt nhất	Vét cạn toàn bộ	Phương án chọn đường đi chính xác có lời giải trực tiếp
Trường hợp xấu nhất	Vét cạn	Vét cạn

CHƯƠNG 2: XÂY DỰNG CHƯƠNG TRÌNH

2.1 Mô tả bài toán

- Mô tả : Cho mê cung có 1 lối vào được kí hiệu là S và một lối ra ký hiệu là E. Làm thế nào để có thể tìm được lối ra
- Toán tử :
 - + Đi sang trái
 - + Đi sang phải
 - + Đi xuống dưới

2.1.1 Không gian trạng thái

Trong trường hợp này không gian trạng thái bài toán tìm đường đi trong mê cung sẽ được xác định như sau :

$T = \{ (x, y) , x \text{ khác } m , y \text{ khác } n \}$

$S = (0, 0)$: trạng thái ban đầu và phụ thuộc vào lối vào I

$G = (m, n)$: trạng thái đích vì chỉ có 1 lối ra nên G chỉ có 1 tập con là (m, n)

F = Tập hợp các thao tác di chuyển sang trái, sang phải, đi xuống để có thể thay đổi vị trí của nhân vật

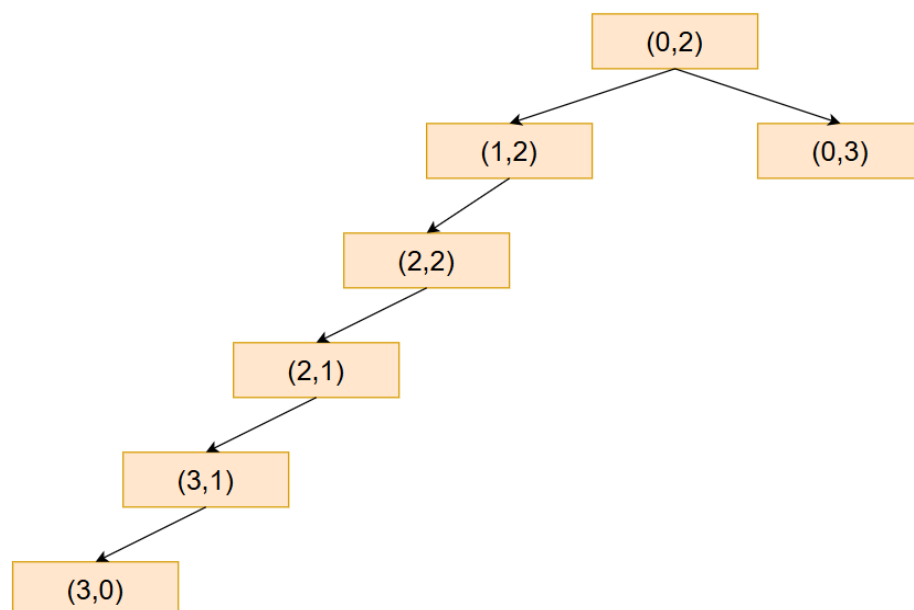
Mô tả không gian trạng thái của bài toán cụ thể với mê cung cho trước như sau :

Mê cung : ++S0

0+0+

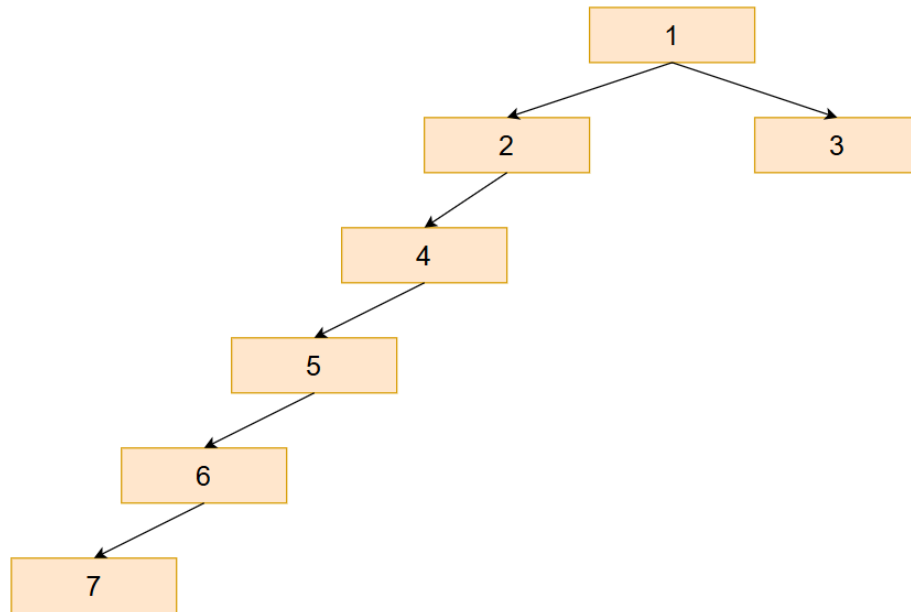
+00+

E0++



Hình 2.1: Mô tả không gian trạng thái

2.1.2 Chuyển đổi không gian trạng thái thành đồ thị



Hình 2.2: Chuyển đổi không gian trạng thái thành đồ thị

2.2 Cài đặt thuật toán

2.2.1 Hàm khởi tạo màn hình

```
# Cài đặt cửa sổ đồ họa
wn = turtle.Screen() # Tạo cửa sổ đồ họa.
wn.bgcolor("black") # Đặt màu nền cửa sổ là đen.
wn.title("A BFS Maze Solving Program") # Đặt tiêu đề cửa sổ.
wn.setup(1300, 700) # Đặt kích thước cửa sổ đồ họa.
```

2.2.2 Lớp vẽ mê cung

```
# Định nghĩa lớp Mê cung (Maze)
class Maze(turtle.Turtle):
    def __init__(self):
        turtle.Turtle.__init__(self)
        self.shape("square") # Định dạng con trỏ hình vuông.
        self.color("white") # Màu sắc của con trỏ là trắng.
        self.penup() # Nhấc bút lên để không vẽ đường.
        self.speed(0) # Tốc độ cao nhất.
```

2.2.3 Lớp green để đánh dấu điểm kết thúc

```
# Định nghĩa lớp cho điểm kết thúc
class Green(turtle.Turtle):
    def __init__(self):
        turtle.Turtle.__init__(self)
        self.shape("square")
        self.color("green") # Màu xanh cho điểm kết thúc.
        self.penup()
        self.speed(0)
```

2.2.4 Lớp red để đánh dấu điểm bắt đầu

```
# Định nghĩa lớp cho điểm bắt đầu
class Red(turtle.Turtle):
    def __init__(self):
        turtle.Turtle.__init__(self)
        self.shape("square")
        self.color("red") # Màu đỏ cho điểm bắt đầu.
        self.penup()
        self.speed(0)
```

2.2.5 Lớp yellow biểu diễn đường đi cuối cùng

```
# Định nghĩa lớp cho đường đi cuối cùng
class Yellow(turtle.Turtle):
    def __init__(self):
        turtle.Turtle.__init__(self)
        self.shape("square")
        self.color("yellow") # Màu vàng cho đường đi kết quả.
        self.penup()
        self.speed(0)
```

2.2.6 Hàm thiết lập mê cung

```
# Hàm thiết lập mê cung
def setup_maze(grid):
    global start_x, start_y, end_x, end_y # Biến toàn cục để lưu tọa độ điểm bắt đầu và kết thúc.
    for y in range(len(grid)): # Duyệt từng dòng của mê cung.
        for x in range(len(grid[y])): # Duyệt từng ký tự trong dòng.
            character = grid[y][x] # Lấy ký tự tại vị trí (x, y).
            screen_x = -588 + (x * 24) # Tính tọa độ x trên màn hình.
            screen_y = 288 - (y * 24) # Tính tọa độ y trên màn hình.

            if character == "+": # Nếu là tường:
                maze.goto(screen_x, screen_y) # Di chuyển con trỏ đến vị trí tương ứng.
                maze.stamp() # Đóng dấu tường.
                walls.append((screen_x, screen_y)) # Lưu tọa độ vào danh sách walls.

            if character == " " or character == "e": # Nếu là đường đi hoặc điểm kết thúc:
                path.append((screen_x, screen_y)) # Lưu vào danh sách path.

            if character == "e": # Nếu là điểm kết thúc:
                green.color("purple")
                green.goto(screen_x, screen_y) # Di chuyển đến điểm kết thúc.
                end_x, end_y = screen_x, screen_y # Lưu tọa độ vào biến toàn cục.
                green.stamp()
                green.color("green")

            if character == "s": # Nếu là điểm bắt đầu:
                start_x, start_y = screen_x, screen_y # Lưu tọa độ vào biến toàn cục.
                red.goto(screen_x, screen_y) # Đặt con trỏ đỏ tại điểm bắt đầu.
```

2.2.7 Hàm thiết lập đường đi ngắn nhất khi thuật toán tìm kiếm kết thúc

```
# Hàm hiển thị đường đi cuối cùng
def backRoute(x, y):
    yellow.goto(x, y) # Di chuyển con trỏ vàng đến tọa độ đích (kết thúc).
    yellow.stamp() # Đánh dấu tọa độ đó bằng màu vàng.
    while (x, y) != (start_x, start_y): # Lặp lại cho đến khi quay về điểm bắt đầu.
        yellow.goto(solution[x, y]) # Di chuyển đến tọa độ trước đó theo từ điển `solution`.
        yellow.stamp() # Đánh dấu ô này bằng màu vàng.
        x, y = solution[x, y] # Cập nhật vị trí hiện tại để quay ngược lại đường đi.
```


2.2.8 Thuật toán BFS (tìm kiếm chiều sâu)

```
# Hàm tìm kiếm BFS
def search(x, y):
    frontier.append((x, y)) # Thêm điểm bắt đầu vào hàng đợi.
    solution[x, y] = x, y # Ghi nhận điểm đầu tiên.

    while len(frontier) > 0: # Lặp cho đến khi hàng đợi rỗng.
        x, y = frontier.popleft() # Lấy phần tử đầu hàng đợi.

        # Kiểm tra các ô kề:
        for dx, dy in [(-24, 0), (0, -24), (24, 0), (0, 24)]:
            neighbor = (x + dx, y + dy) # Tạo tọa độ ô liền kề từ vị trí hiện tại.
            if neighbor in path and neighbor not in visited:
                # Kiểm tra xem ô này có thuộc đường đi hợp lệ và chưa được thăm hay không.
                frontier.append(neighbor) # Thêm ô liền kề vào hàng đợi để duyệt tiếp.
                solution[neighbor] = (x, y) # Lưu lại đường đi theo cơ chế backtracking.
                visited.add(neighbor) # Đánh dấu ô này đã được thăm.
                green.goto(neighbor) # Di chuyển con trỏ xanh lá đến ô này.
                green.stamp() # Đóng dấu để đánh dấu ô vừa duyệt.
```

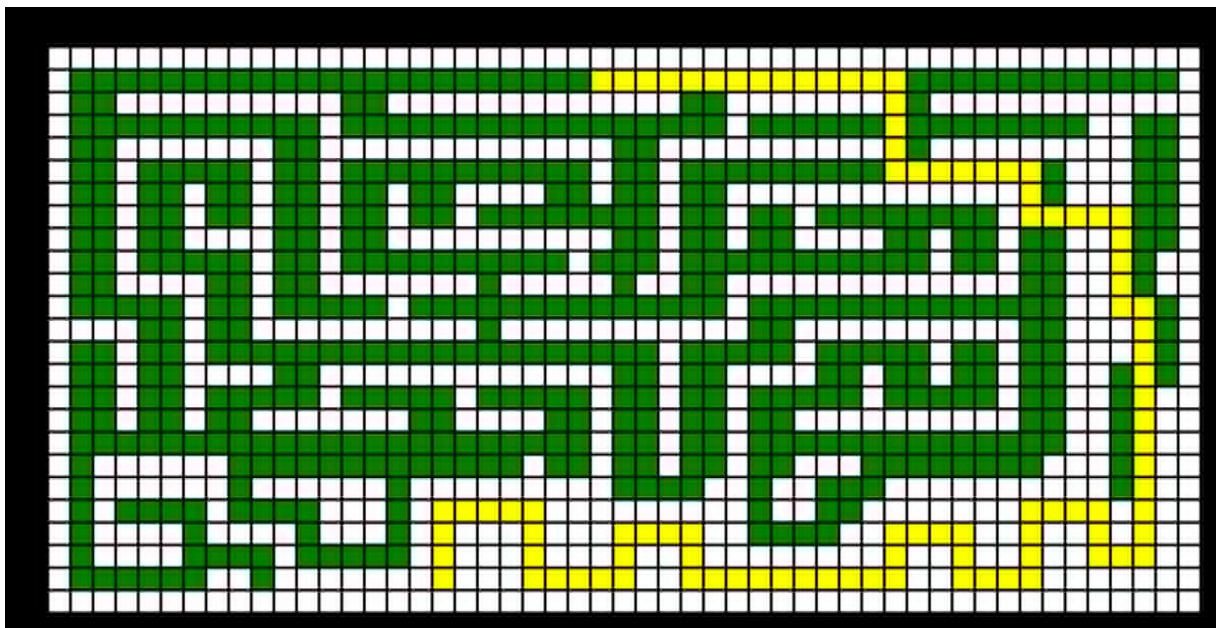
2.2.9 Thiết lập cuối cùng trước khi test

```
# Khởi tạo các lớp và danh sách:
maze = Maze() # Tạo đối tượng mê cung.
red = Red() # Tạo đối tượng biểu diễn điểm bắt đầu.
green = Green() # Tạo đối tượng biểu diễn điểm kết thúc.
yellow = Yellow() # Tạo đối tượng biểu diễn đường đi cuối cùng.

walls = [] # Danh sách lưu tọa độ của các bức tường trong mê cung.
path = [] # Danh sách lưu tọa độ của các ô đường đi hợp lệ.
visited = set() # Tập hợp các ô đã được thăm.
frontier = deque() # Hàng đợi dùng cho thuật toán BFS.
solution = {} # Từ điển để lưu thông tin đường đi (backtracking).

# Chương trình chính:
setup_maze(grid) # Thiết lập mê cung bằng cách đọc lưới (grid) và phân loại tường, đường đi.
search(start_x, start_y) # Tìm kiếm đường đi từ điểm bắt đầu đến điểm kết thúc bằng BFS.
backRoute(end_x, end_y) # Hiển thị đường đi cuối cùng từ điểm kết thúc quay về điểm bắt đầu.
wn.exitonclick() # Chờ người dùng nhấp chuột để đóng cửa sổ đồ họa.
```

2.3 Demo chương trình



Tư tưởng của chương trình:

- Hàm setup_maze(grid)

+ Chức năng:

- Thiết lập và hiển thị mê cung trên màn hình dựa trên grid được cung cấp.

+ Hoạt động:

- Ý nghĩa các tham số và biến
 - grid: Là danh sách các chuỗi, mỗi ký tự trong chuỗi đại diện cho một phần tử của mê cung (+, s, e, hoặc).
 - start_x, start_y: Tọa độ bắt đầu của mê cung.
 - end_x, end_y: Tọa độ kết thúc của mê cung.

+ Quy trình:

- Duyệt từng dòng (y) và từng ký tự trong dòng (x) của grid.
- Tính toán tọa độ trên màn hình cho từng ô trong grid bằng công thức:
 - $screen_x = -588 + (x * 24)$ (vị trí ngang).
 - $screen_y = 288 - (y * 24)$ (vị trí dọc).
- Tùy thuộc vào ký tự tại vị trí (x, y):
 - +: Hiển thị tường và thêm tọa độ vào danh sách walls.
 - hoặc e: Đánh dấu là đường đi (path).
 - e: Đặt điểm kết thúc (green) tại vị trí đó.
 - s: Đặt điểm bắt đầu (red) tại vị trí đó.

- endProgram()

+ Chức năng:

- Đóng cửa sổ đồ họa và kết thúc chương trình.

+ Hoạt động:

- Gọi hàm `wn.exitonclick()` để chờ người dùng nhấp chuột trước khi thoát.
- Dùng `sys.exit()` để thoát hoàn toàn chương trình.

- search (x, y)

+ Chức năng:

- Thực hiện thuật toán BFS (Breadth-First Search) để tìm đường đi từ điểm bắt đầu đến điểm kết thúc.

+ Hoạt động:

• Khởi tạo:

- Đưa điểm bắt đầu (x, y) vào hàng đợi frontier.
- Lưu điểm bắt đầu vào từ điển solution (đánh dấu không có điểm trước đó).

- **Lắp qua hàng đợi:**

- Lấy điểm tiếp theo từ frontier (x, y).
- Kiểm tra các ô liền kề (trái, phải, trên, dưới):
Nếu là đường đi hợp lệ và chưa được duyệt:
Thêm ô đó vào solution (lưu lại ô trước đó để phục vụ việc backtrack).
Đưa vào hàng đợi frontier.
Đánh dấu đã duyệt (visited).

- **Hiển thị trực quan:**

- Dùng green để đánh dấu các điểm mà BFS đã ghé thăm.

- **backRoute (x,y)**

- + **Chức năng:**

- Dùng từ điển solution để truy vết đường đi ngắn nhất từ điểm kết thúc (e) về điểm bắt đầu (s).

- + **Hoạt động:**

- Khởi tạo:
 - Đưa yellow đến vị trí kết thúc (x, y).
- Duyệt ngược:
 - Dùng solution để tìm tọa độ trước đó của ô hiện tại.
 - Cập nhật tọa độ mới cho yellow và hiển thị.
 - Tiếp tục cho đến khi yellow đạt đến điểm bắt đầu (start_x, start_y).
- **Maze:** Tạo các khối màu trắng đại diện cho tường.
- **Green:** Tạo khối màu xanh lá cây để đánh dấu điểm kết thúc.
- **Red:** Tạo khối màu đỏ để đánh dấu điểm bắt đầu.
- **Yellow:** Hiển thị đường đi ngắn nhất từ điểm bắt đầu đến điểm kết thúc.

KẾT LUẬN

Qua quá trình nghiên cứu và thực hiện bài tập lớn với đề tài tìm đường đi trong mê cung bằng các thuật toán tìm kiếm mù, chúng em đã có cơ hội tiếp cận, áp dụng và hiểu rõ hơn về những kiến thức đã học trong môn “Trí tuệ nhân tạo”. Việc triển khai các thuật toán như BFS, DFS trên một bài toán cụ thể không chỉ giúp chúng em củng cố nền tảng lý thuyết mà còn rèn luyện tư duy phân tích, kỹ năng lập trình và đánh giá hiệu quả giải pháp.

Chúng em nhận thấy rằng, mặc dù là những thuật toán cơ bản, nhưng khi được áp dụng đúng cách, chúng vẫn mang lại giá trị thực tiễn cao và đóng vai trò quan trọng trong việc giải quyết nhiều bài toán trong lĩnh vực AI. Bên cạnh đó, những khó khăn gặp phải trong quá trình thực hiện cũng là động lực để chúng em không ngừng trau dồi kiến thức và nâng cao năng lực bản thân.

Chúng em xin chân thành cảm ơn sự hướng dẫn tận tâm của cô Mai Thanh Hồng. Những kiến thức và kinh nghiệm thu được từ bài tập lớn này sẽ là hành trang quý báu giúp chúng em tự tin hơn trên con đường học tập và phát triển trong lĩnh vực trí tuệ nhân tạo nói riêng và công nghệ thông tin nói chung.

TÀI LIỆU THAM KHẢO

- [1] Nguyễn Phương Nga, Trần Hùng Cường, *Giáo trình Trí tuệ nhân tạo*, NXB Thống kê, 2021.
- [2] Nguyễn Thanh Thủy, *Trí tuệ nhân tạo*, NXB Thống kê, 1999.
- [3] Nguyễn Đình Thức, *Giáo trình Trí tuệ nhân tạo: Mạng neuron phương pháp và ứng dụng*, NXB Giáo dục, 2000.

PHIẾU HỌC TẬP CÁ NHÂN

I. Thông tin chung:

1. Tên lớp: 20242IT6094003 Khoa: 18
2. Họ và tên sinh viên: Trần Nguyễn Trà Giang - 2023600740

II. Nội dung học tập:

1. Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung
2. Hoạt động của sinh viên:

- Hoạt động 1: Đề xuất chủ đề nghiên cứu

+ Nội dung:

- Đề xuất lựa chọn chủ đề nghiên cứu và xin ý kiến người hướng dẫn về chủ đề nghiên cứu
- Lập biên bản họp và làm việc nhóm tuần 1 và tuần 5 (theo BM03)
- Tham gia đặt ra các quy tắc làm việc nhóm

+ Mục tiêu/chuẩn đầu ra: Biết cách lập kế hoạch thực hiện bài tập lớn (theo BM02) và lập biên bản họp và làm việc nhóm (theo BM03)

- Hoạt động 2: Viết đề cương nghiên cứu

+ Nội dung:

- Tham gia viết đề cương nghiên cứu
- Bảo vệ đề cương nghiên cứu

+ Mục tiêu/chuẩn đầu ra: Viết được đề cương nghiên cứu

- Hoạt động 3: Báo cáo tiến độ lần 1

Mục tiêu/chuẩn đầu ra: Trình bày được thuật toán tìm kiếm mù theo chiều rộng và thuật toán AT

- Hoạt động 4: Báo cáo tiến độ lần 2

Mục tiêu/chuẩn đầu ra: Tham gia mô tả không gian trạng thái của bài toán được chọn và chuyển đổi không gian trạng thái thành đồ thị. Sử dụng ngôn ngữ lập trình Python để ứng dụng thuật toán tìm kiếm mù giải bài toán tìm đường đi ngắn nhất trong mê cung

3. Sản phẩm nghiên cứu: Quyển báo cáo thí nghiệm/thực nghiệm và chương trình code giải quyết bài toán tìm đường đi ngắn nhất trong mê cung

III. Nhiệm vụ học tập:

1. Hoàn thành Tiểu luận, Bài tập lớn, Đồ án/Dự án theo đúng thời gian quy định (từ ngày 09/04/2025 đến ngày 26/05/2025)
2. Báo cáo sản phẩm nghiên cứu theo chủ đề được giao trước giảng viên và những sinh viên khác

IV. Học liệu thực hiện Bài tập lớn:

1. Tài liệu học tập:

[1] Nguyễn Phương Nga, Trần Hùng Cường, *Giáo trình Trí tuệ nhân tạo*, NXB Thống kê, 2021.

[2] Nguyễn Thanh Thủy, *Trí tuệ nhân tạo*, NXB Thống kê, 1999.

[3] Nguyễn Đình Thúc, *Giáo trình Trí tuệ nhân tạo: Mạng neuron phương pháp và ứng dụng*, NXB Giáo dục, 2000.

2. Phương tiện, nguyên liệu thực hiện Bài tập lớn: Máy tính cá nhân, máy chiếu, mạng internet.

BÁO CÁO HỌC TẬP CÁ NHÂN

Tên lớp: 20242IT6094003

Khoá: 18

Họ và tên sinh viên: Trần Nguyễn Trà Giang - 2023600740

Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung

Tuần	Người thực hiện	Nội dung công việc	Kết quả đạt được	Kiến nghị với giảng viên hướng dẫn (Nêu những khó khăn, hỗ trợ từ phía giảng viên, ... nếu cần)
1	Trần Nguyễn Trà Giang	Trình bày về thuật toán tìm kiếm mù theo chiều rộng	Hoàn thành đúng thời hạn và đầy đủ nội dung	
2	Trần Nguyễn Trà Giang	Trình bày về thuật toán AT	Hoàn thành đúng thời hạn và đầy đủ nội dung	
3	Trần Nguyễn Trà Giang	Mô tả không gian trạng thái và chuyển đổi không gian trạng thái thành đồ thị	Hoàn thành đúng thời hạn và đầy đủ nội dung	
4	Trần Nguyễn Trà Giang	Cài đặt chương trình và chạy được chương trình	Hoàn thành đúng thời hạn và đầy đủ nội dung	
5	Trần Nguyễn Trà Giang	Tham gia hoàn thiện báo cáo và các biểu mẫu, phiếu liên quan	Hoàn thành đúng thời hạn và đầy đủ nội dung	

Ngày tháng năm 2025

XÁC NHẬN CỦA GIẢNG VIÊN

(Ký, ghi rõ họ tên)

**KẾ HOẠCH THỰC HIỆN TIỂU LUẬN, BÀI TẬP LỚN, ĐỒ
ÁN/DỰ ÁN**

Tên lớp: 20242IT6094003

Khoá: 18

Họ và tên sinh viên: Trần Nguyễn Trà Giang - 2023600740

Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung

Tuần	Người thực hiện	Nội dung công việc	Phương pháp thực hiện
1	Trần Nguyễn Trà Giang	Trình bày về thuật toán tìm kiếm mù theo chiều rộng	Nghiên cứu, tìm hiểu, áp dụng những phần đã học và tra cứu tài liệu tham khảo
2	Trần Nguyễn Trà Giang	Trình bày về thuật toán AT	Nghiên cứu, tìm hiểu, áp dụng những phần đã học và tra cứu tài liệu tham khảo
3	Trần Nguyễn Trà Giang	Mô tả không gian trạng thái của bài toán và chuyển đổi không gian trạng thái thành đồ thị	Nghiên cứu, tìm hiểu và tra cứu tài liệu tham khảo
4	Trần Nguyễn Trà Giang	Cài đặt chương trình và chạy được chương trình	Nghiên cứu, tìm hiểu, áp dụng những phần đã học và tra cứu tài liệu tham khảo
5	Trần Nguyễn Trà Giang	Test code của bài toán, hoàn thiện báo cáo và các biểu mẫu, phiếu liên quan	Tổng hợp lại tất cả các nội dung nhóm đã làm

Ngày tháng năm 2025

XÁC NHẬN CỦA GIẢNG VIÊN
(Ký, ghi rõ họ tên)

PHIẾU HỌC TẬP CÁ NHÂN

I. Thông tin chung:

1. Tên lớp: 20242IT6094003 Khoá: 18
2. Họ và tên sinh viên: Hoàng Xuân Hưng - 2023601662

II. Nội dung học tập:

1. Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung
2. Hoạt động của sinh viên:

- Hoạt động 1: Đề xuất chủ đề nghiên cứu

+ Nội dung:

- Đề xuất lựa chọn chủ đề nghiên cứu và xin ý kiến người hướng dẫn về chủ đề nghiên cứu
- Lập biên bản họp và làm việc nhóm tuần 2 (theo BM03)
- Tham gia đặt ra các quy tắc làm việc nhóm

+ Mục tiêu/chuẩn đầu ra: Biết cách lập kế hoạch thực hiện bài tập lớn (theo BM02) và lập biên bản họp và làm việc nhóm (theo BM03)

- Hoạt động 2: Viết đề cương nghiên cứu

+ Nội dung:

- Tham gia viết đề cương nghiên cứu
- Bảo vệ đề cương nghiên cứu

+ Mục tiêu/chuẩn đầu ra: Viết được đề cương nghiên cứu

- Hoạt động 3: Báo cáo tiến độ lần 1

Mục tiêu/chuẩn đầu ra: Trình bày về không gian trạng thái và thuật toán tối ưu

- Hoạt động 4: Báo cáo tiến độ lần 2

Mục tiêu/chuẩn đầu ra: Tham gia mô tả không gian trạng thái của bài toán được chọn và chuyển đổi không gian trạng thái thành đồ thị. Sử dụng ngôn ngữ lập trình Python để ứng dụng thuật toán tìm kiếm mù giải bài toán tìm đường đi ngắn nhất trong mê cung

3. Sản phẩm nghiên cứu: Quyển báo cáo thí nghiệm/thực nghiệm và chương trình code giải quyết bài toán tìm đường đi ngắn nhất trong mê cung

III. Nhiệm vụ học tập:

1. Hoàn thành Tiểu luận, Bài tập lớn, Đồ án/Dự án theo đúng thời gian quy định (từ ngày 09/04/2025 đến ngày 26/05/2025)
2. Báo cáo sản phẩm nghiên cứu theo chủ đề được giao trước giảng viên và những sinh viên khác

IV. Học liệu thực hiện Bài tập lớn:

1. Tài liệu học tập:

[1] Nguyễn Phương Nga, Trần Hùng Cường, *Giáo trình Trí tuệ nhân tạo*, NXB Thống kê, 2021.

[2] Nguyễn Thanh Thủy, *Trí tuệ nhân tạo*, NXB Thống kê, 1999.

[3] Nguyễn Đình Thúc, *Giáo trình Trí tuệ nhân tạo: Mạng neuron phương pháp và ứng dụng*, NXB Giáo dục, 2000.

2. Phương tiện, nguyên liệu thực hiện Bài tập lớn: Máy tính cá nhân, máy chiếu, mạng internet.

BÁO CÁO HỌC TẬP CÁ NHÂN

Tên lớp: 20242IT6094003

Khoá: 18

Họ và tên sinh viên: Hoàng Xuân Hưng - 2023601662

Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung

Tuần	Người thực hiện	Nội dung công việc	Kết quả đạt được	Kiến nghị với giảng viên hướng dẫn (Nêu những khó khăn, hỗ trợ từ phía giảng viên, ... nếu cần)
1	Hoàng Xuân Hưng	Trình bày về không gian trạng thái	Hoàn thành đúng thời hạn và đầy đủ nội dung	
2	Hoàng Xuân Hưng	Trình bày về thuật toán tối ưu	Hoàn thành đúng thời hạn và đầy đủ nội dung	
3	Hoàng Xuân Hưng	Mô tả không gian trạng thái và chuyển đổi không gian trạng thái thành đồ thị	Hoàn thành đúng thời hạn và đầy đủ nội dung	
4	Hoàng Xuân Hưng	Cài đặt chương trình và chạy được chương trình	Hoàn thành đúng thời hạn và đầy đủ nội dung	
5	Hoàng Xuân Hưng	Tham gia hoàn thiện báo cáo và các biểu mẫu, phiếu liên quan	Hoàn thành đúng thời hạn và đầy đủ nội dung	

Ngày tháng năm 2025

XÁC NHẬN CỦA GIẢNG VIÊN

(Ký, ghi rõ họ tên)

**KẾ HOẠCH THỰC HIỆN TIỂU LUẬN, BÀI TẬP LỚN, ĐỒ
ÁN/DỰ ÁN**

Tên lớp: 20242IT6094003

Khoá: 18

Họ và tên sinh viên: Hoàng Xuân Hưng - 2023601662

Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung

Tuần	Người thực hiện	Nội dung công việc	Phương pháp thực hiện
1	Hoàng Xuân Hưng	Trình bày về không gian trạng thái	Nghiên cứu, tìm hiểu, áp dụng những phần đã học và tra cứu tài liệu tham khảo
2	Hoàng Xuân Hưng	Trình bày về thuật toán tối ưu	Nghiên cứu, tìm hiểu, áp dụng những phần đã học và tra cứu tài liệu tham khảo
3	Hoàng Xuân Hưng	Mô tả không gian trạng thái của bài toán và chuyển đổi không gian trạng thái thành đồ thị	Nghiên cứu, tìm hiểu và tra cứu tài liệu tham khảo
4	Hoàng Xuân Hưng	Cài đặt chương trình và chạy được chương trình	Nghiên cứu, tìm hiểu, áp dụng những phần đã học và tra cứu tài liệu tham khảo
5	Hoàng Xuân Hưng	Test code của bài toán, hoàn thiện báo cáo và các biểu mẫu, phiếu liên quan	Tổng hợp lại tất cả các nội dung nhóm đã làm

Ngày tháng năm 2025

XÁC NHẬN CỦA GIẢNG VIÊN
(Ký, ghi rõ họ tên)

PHIẾU HỌC TẬP CÁ NHÂN

I. Thông tin chung:

1. Tên lớp: 20242IT6094003 Khoá: 18
2. Họ và tên sinh viên: Tạ Tuấn Nguyên - 2023601547

II. Nội dung học tập:

1. Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung
2. Hoạt động của sinh viên:

- Hoạt động 1: Đề xuất chủ đề nghiên cứu

+ Nội dung:

- Đề xuất lựa chọn chủ đề nghiên cứu và xin ý kiến người hướng dẫn về chủ đề nghiên cứu
- Lập biên bản họp và làm việc nhóm tuần 3 (theo BM03)
- Tham gia đặt ra các quy tắc làm việc nhóm

+ Mục tiêu/chuẩn đầu ra: Biết cách lập kế hoạch thực hiện bài tập lớn (theo BM02) và lập biên bản họp và làm việc nhóm (theo BM03)

- Hoạt động 2: Viết đề cương nghiên cứu

+ Nội dung:

- Tham gia viết đề cương nghiên cứu
- Bảo vệ đề cương nghiên cứu

+ Mục tiêu/chuẩn đầu ra: Viết được đề cương nghiên cứu

- Hoạt động 3: Báo cáo tiến độ lần 1

Mục tiêu/chuẩn đầu ra: Viết được lời mở đầu, lời cảm ơn và trình bày được thuật toán AKT

- Hoạt động 4: Báo cáo tiến độ lần 2

Mục tiêu/chuẩn đầu ra: Tham gia mô tả không gian trạng thái của bài toán được chọn và chuyển đổi không gian trạng thái thành đồ thị. Sử dụng ngôn ngữ lập trình Python để ứng dụng thuật toán tìm kiếm mù giải bài toán tìm đường đi ngắn nhất trong mê cung

3. Sản phẩm nghiên cứu: Quyển báo cáo thí nghiệm/thực nghiệm và chương trình code giải quyết bài toán tìm đường đi ngắn nhất trong mê cung

III. Nhiệm vụ học tập:

1. Hoàn thành Tiểu luận, Bài tập lớn, Đồ án/Dự án theo đúng thời gian quy định (từ ngày 09/04/2025 đến ngày 26/05/2025)
2. Báo cáo sản phẩm nghiên cứu theo chủ đề được giao trước giảng viên và những sinh viên khác

IV. Học liệu thực hiện Bài tập lớn:

1. Tài liệu học tập:

[1] Nguyễn Phương Nga, Trần Hùng Cường, *Giáo trình Trí tuệ nhân tạo*, NXB Thống kê, 2021.

[2] Nguyễn Thanh Thủy, *Trí tuệ nhân tạo*, NXB Thống kê, 1999.

[3] Nguyễn Đình Thúc, *Giáo trình Trí tuệ nhân tạo: Mạng neuron phương pháp và ứng dụng*, NXB Giáo dục, 2000.

2. Phương tiện, nguyên liệu thực hiện Bài tập lớn: Máy tính cá nhân, máy chiếu, mạng internet.

BÁO CÁO HỌC TẬP CÁ NHÂN

Tên lớp: 20242IT6094003

Khoá: 18

Họ và tên sinh viên: Tạ Tuấn Nguyên – 2023601547

Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung

Tuần	Người thực hiện	Nội dung công việc	Kết quả đạt được	Kiến nghị với giảng viên hướng dẫn (Nêu những khó khăn, hỗ trợ từ phía giảng viên,... nếu cần)
1	Tạ Tuấn Nguyên	Viết lời mở đầu và lời cảm ơn	Hoàn thành đúng thời hạn và đầy đủ nội dung	
2	Tạ Tuấn Nguyên	Trình bày về thuật toán AKT	Hoàn thành đúng thời hạn và đầy đủ nội dung	
3	Tạ Tuấn Nguyên	Mô tả không gian trạng thái và chuyển đổi không gian trạng thái thành đồ thị	Hoàn thành đúng thời hạn và đầy đủ nội dung	
4	Tạ Tuấn Nguyên	Cài đặt chương trình và chạy được chương trình	Hoàn thành đúng thời hạn và đầy đủ nội dung	
5	Tạ Tuấn Nguyên	Tham gia hoàn thiện báo cáo và các biểu mẫu, phiếu liên quan	Hoàn thành đúng thời hạn và đầy đủ nội dung	

Ngày tháng năm 2025
XÁC NHẬN CỦA GIẢNG VIÊN
(Ký, ghi rõ họ tên)

**KẾ HOẠCH THỰC HIỆN TIỂU LUẬN, BÀI TẬP LỚN, ĐỒ
ÁN/DỰ ÁN**

Tên lớp: 20242IT6094003

Khoá: 18

Họ và tên sinh viên: Tạ Tuấn Nguyên - 2023601547

Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung

Tuần	Người thực hiện	Nội dung công việc	Phương pháp thực hiện
1	Tạ Tuấn Nguyên	Viết lời mở đầu và lời cảm ơn	Nghiên cứu, tìm hiểu, áp dụng những phần đã học và tra cứu tài liệu tham khảo
2	Tạ Tuấn Nguyên	Trình bày về thuật toán AKT	Nghiên cứu, tìm hiểu, áp dụng những phần đã học và tra cứu tài liệu tham khảo
3	Tạ Tuấn Nguyên	Mô tả không gian trạng thái của bài toán và chuyển đổi không gian trạng thái thành đồ thị	Nghiên cứu, tìm hiểu và tra cứu tài liệu tham khảo
4	Tạ Tuấn Nguyên	Cài đặt chương trình và chạy được chương trình	Nghiên cứu, tìm hiểu, áp dụng những phần đã học và tra cứu tài liệu tham khảo
5	Tạ Tuấn Nguyên	Test code của bài toán, hoàn thiện báo cáo và các biểu mẫu, phiếu liên quan	Tổng hợp lại tất cả các nội dung nhóm đã làm

Ngày tháng năm 2025

XÁC NHẬN CỦA GIẢNG VIÊN
(Ký, ghi rõ họ tên)

PHIẾU HỌC TẬP CÁ NHÂN

I. Thông tin chung:

1. Tên lớp: 2024IT6094003 Khoa: 18
2. Họ và tên sinh viên: Phùng Bảo Trường-2023600462

II. Nội dung học tập:

1. Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung
2. Hoạt động của sinh viên:

- Hoạt động 1: Đề xuất chủ đề nghiên cứu

+ Nội dung:

- Đề xuất lựa chọn chủ đề nghiên cứu và xin ý kiến người hướng dẫn về chủ đề nghiên cứu
- Lập biên bản họp và làm việc nhóm tuần 4 (theo BM03)
- Tham gia đặt ra các quy tắc làm việc nhóm

+ Mục tiêu/chuẩn đầu ra: Biết cách lập kế hoạch thực hiện bài tập lớn (theo BM02) và lập biên bản họp và làm việc nhóm (theo BM03)

- Hoạt động 2: Viết đề cương nghiên cứu

+ Nội dung:

- Tham gia viết đề cương nghiên cứu
- Bảo vệ đề cương nghiên cứu

+ Mục tiêu/chuẩn đầu ra: Viết được đề cương nghiên cứu

- Hoạt động 3: Báo cáo tiến độ lần 1

Mục tiêu/chuẩn đầu ra: Trình bày về thuật toán tìm kiếm theo chiều sâu và thuật toán A*

- Hoạt động 4: Báo cáo tiến độ lần 2

Mục tiêu/chuẩn đầu ra: Tham gia mô tả không gian trạng thái của bài toán được chọn và chuyển đổi không gian trạng thái thành đồ thị. Sử dụng ngôn ngữ lập trình Python để ứng dụng thuật toán tìm kiếm mù giải bài toán tìm đường đi ngắn nhất trong mê cung

3. Sản phẩm nghiên cứu: Quyển báo cáo thí nghiệm/thực nghiệm và chương trình code giải quyết bài toán tìm đường đi ngắn nhất trong mê cung

III. Nhiệm vụ học tập:

1. Hoàn thành Tiểu luận, Bài tập lớn, Đồ án/Dự án theo đúng thời gian quy định (từ ngày 09/04/2025 đến ngày 26/05/2025)
2. Báo cáo sản phẩm nghiên cứu theo chủ đề được giao trước giảng viên và những sinh viên khác

IV. Học liệu thực hiện Bài tập lớn:

1. Tài liệu học tập:

[1] Nguyễn Phương Nga, Trần Hùng Cường, *Giáo trình Trí tuệ nhân tạo*, NXB Thống kê, 2021.

[2] Nguyễn Thanh Thủy, *Trí tuệ nhân tạo*, NXB Thống kê, 1999.

[3] Nguyễn Đình Thúc, *Giáo trình Trí tuệ nhân tạo: Mạng neuron phương pháp và ứng dụng*, NXB Giáo dục, 2000.

2. Phương tiện, nguyên liệu thực hiện Bài tập lớn: Máy tính cá nhân, máy chiếu, mạng internet.

BÁO CÁO HỌC TẬP CÁ NHÂN

Tên lớp: 20242IT6094003

Khoá: 18

Họ và tên sinh viên: Phùng Bảo Trường - 2023600462

Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung

Tuần	Người thực hiện	Nội dung công việc	Kết quả đạt được	Kiến nghị với giảng viên hướng dẫn (Nêu những khó khăn, hỗ trợ từ phía giảng viên, ... nếu cần)
1	Phùng Bảo Trường	Trình bày về thuật toán tìm kiếm mù theo chiều sâu	Hoàn thành đúng thời hạn và đầy đủ nội dung	
2	Phùng Bảo Trường	Trình bày về thuật toán A*	Hoàn thành đúng thời hạn và đầy đủ nội dung	
3	Phùng Bảo Trường	Mô tả không gian trạng thái và chuyển đổi không gian trạng thái thành đồ thị	Hoàn thành đúng thời hạn và đầy đủ nội dung	
4	Phùng Bảo Trường	Cài đặt chương trình và chạy được chương trình	Hoàn thành đúng thời hạn và đầy đủ nội dung	
5	Phùng Bảo Trường	Hoàn thiện báo cáo và các biểu mẫu, phiếu liên quan	Hoàn thành đúng thời hạn và đầy đủ nội dung	

Ngày tháng năm 2025

XÁC NHẬN CỦA GIẢNG VIÊN

(Ký, ghi rõ họ tên)

**KẾ HOẠCH THỰC HIỆN TIỂU LUẬN, BÀI TẬP LỚN, ĐỒ
ÁN/DỰ ÁN**

Tên lớp: 20242IT6094003

Khoá: 18

Họ và tên sinh viên: Phùng Bảo Trường - 2023600462

Tên chủ đề: Ứng dụng thuật toán mù để tìm đường đi ngắn nhất trong mê cung

Tuần	Người thực hiện	Nội dung công việc	Phương pháp thực hiện
1	Phùng Bảo Trường	Trình bày về thuật toán tìm kiếm mù theo chiều sâu	Nghiên cứu, tìm hiểu, áp dụng những phần đã học và tra cứu tài liệu tham khảo
2	Phùng Bảo Trường	Trình bày về thuật toán A*	Nghiên cứu, tìm hiểu, áp dụng những phần đã học và tra cứu tài liệu tham khảo
3	Phùng Bảo Trường	Mô tả không gian trạng thái của bài toán và chuyển đổi không gian trạng thái thành đồ thị	Nghiên cứu, tìm hiểu và tra cứu tài liệu tham khảo
4	Phùng Bảo Trường	Cài đặt chương trình và chạy được chương trình	Nghiên cứu, tìm hiểu, áp dụng những phần đã học và tra cứu tài liệu tham khảo
5	Phùng Bảo Trường	Hoàn thiện báo cáo và các biểu mẫu, phiếu liên quan	Tổng hợp lại tất cả các nội dung nhóm đã làm

Ngày tháng năm 2025

XÁC NHẬN CỦA GIẢNG VIÊN

(Ký, ghi rõ họ tên)