

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



BÁO CÁO BÀI TẬP LỚN HỌC PHẦN: TRÍ TUỆ NHÂN TẠO
ỨNG DỤNG THUẬT TOÁN TÌM KIẾM MÙ VÀ ÁP
DỤNG GIẢI BÀI TOÁN SUDOKU (4X4)

Giáo Viên Hướng Dẫn : Mai Thanh Hồng

Lớp : 2024IT6094003

Thành Viên Nhóm : Trần Đình Quang

Phạm Thị Phương

Nguyễn Thị Minh

Nguyễn Duy Thái

Hà Nội, 2025

BÁO CÁO HỌC TẬP NHÓM

Tên lớp: 20242IT6094003

Khoá: 18

Tên nhóm: 17

Họ và tên thành viên trong nhóm:

(1) Họ và tên SV: Trần Đình Quang

Mã SV: 2023607628

(2) Họ và tên SV: Nguyễn Duy Thái

Mã SV: 2023601474

(3) Họ và tên SV: Phạm Thị Phương

Mã SV: 2023606873

(4) Họ và tên SV: Nguyễn Thị Minh

Mã SV: 2023606995

Tên chủ đề: Tìm hiểu thuật toán tìm kiếm mù và áp dụng để giải bài toán Sudoku đơn giản (4x4).

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 mở đầu, lời cảm ơn, tìm hiểu về thuật toán DFS, BFS và trò chơi Sudoku	Các thành viên hoàn thành đúng thời hạn và đầy đủ nội dung	
2	Cả nhóm	Tìm hiểu chi tiết về thuật toán DFS, BFS các nội dung lý thuyết liên quan	Các thành viên hoàn thành đúng thời hạn và đầy đủ nội dung	

		đến đề tài, hoàn thiện nội dung chương 1		
3	Cả nhóm	Tìm hiểu trò chơi Sudoku, mô hình hóa trò chơi thành bài toán tìm kiếm, tìm hiểu thuật toán để ứng dụng thuật toán DFS, BFS vào giải quyết bài toán	Các thành viên hoàn thành đúng thời hạn và đầy đủ nội dung	
4	Cả nhóm	Cài đặt chương trình và chạy được chương trình, hoàn thiện chương 2	Cả nhóm đã thống nhất về các biến và xây dựng thuật toán	
5	Cả nhóm	Hoàn thiện chương 3	Cả nhóm hoàn thành đầy đủ và đúng hạn	
6	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:

Tên lớp: 20242IT6094003

Khoá: 18

Tên nhóm: 17

Họ và tên thành viên trong nhóm:

(1)Họ và tên SV: Trần Đình Quang	Mã SV: 2023607628
(2)Họ và tên SV: Nguyễn Duy Thái	Mã SV: 2023601474
(3)Họ và tên SV: Phạm Thị Phương	Mã SV: 2023606873
(4)Họ và tên SV: Nguyễn Thị Minh	Mã SV: 2023606995

II. Nội dung học tập:

1. Tên chủ đề: **Ứng dụng thuật toán tìm kiếm mù và áp dụng giải bài toán Sudoku đơn giản (4x4).**

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 (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).
- 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 để giải bài toán Sudoku đơn giản (4x4).
- 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: Quyển báo cáo thí nghiệm/thực nghiệm + Chương trình code

III. Nhiệm vụ học tập:

1. Hoàn thành báo cáo thí nghiệm/thực nghiệm theo đúng thời gian quy định (từ ngày 09/04/2025 đến ngày 16/5/2025)
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: 17								
3. Ngày bắt đầu: 09/04/2025								
4. Ngày kết thúc: 16/5/2025								
5. Thành viên nhóm: <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div>(1)Họ và tên SV: Trần Đình Quang</div> <div>Mã SV: 2023607628</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div>(2)Họ và tên SV: Nguyễn Duy Thái</div> <div>Mã SV: 2023601474</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div>(3)Họ và tên SV: Phạm Thị Phương</div> <div>Mã SV: 2023606873</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div>(4)Họ và tên SV: Nguyễn Thị Minh</div> <div>Mã SV: 2023606995</div> </div>								
#	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	09/04/2025	09/04/2025	13/04/2025	13/04/2025	Done	Phạm Thị Phương	

	và lời cảm ơn							
2	Trình bày về không gian trạng thái	09/04 /2025	09/04 /2025	13/04 /2025	13/0 4/20 25	Done	Nguyễn Thị Minh	
3	Trình bày về thuật toán tìm kiếm mù	09/04 /2025	09/04 /2025	13/04 /2025	13/0 4/20 25	Done	Nguyễn Duy Thái, Trần Đình Quang	
4	Mô tả không gian trạng thái bài toán	23/04 /2025	23/04 /2025	26/04 /2025	26/0 4/20 25	Done	Nguyễn Duy Thái, Trần Đình Quang, Phạm Thị Phương, Nguyễn Thị Minh	
5	Chuyển đổi không	23/04 /2025	23/04 /2025	26/04 /2025	26/0 4/20 25	Done	Nguyễn Duy Thái, Trần Đình Quang,	

	gian trạng thái thành đồ thị						Phạm Thị Phương, Nguyễn Thị Minh	
6	Cài đặt chương trình bằng thuật toán tìm kiếm BFS và DFS	28/04 /2025	28/04 /2025	02/05 /2025	02/0 5/20 25	Done	Nguyễn Duy Thái, Trần Đình Quang, Phạm Thị Phương, Nguyễn Thị Minh	
7	Thử nghiệm và chạy thử chương trình	28/04 /2025	28/04 /2025	02/05 /2025	02/0 5/20 25	Done	Nguyễn Duy Thái, Trần Đình Quang, Phạm Thị Phương, Nguyễn Thị Minh	
8	Hoàn thiện báo cáo và các	06/05 /2025	06/05 /2025	16/05 /2025	16/0 5/20 25	Done	Nguyễn Duy Thái, Trần Đình Quang,	

	biểu mẫu, phiếu liên quan						Phạm Thị Phương, Nguyễn Thị Minh	
--	---------------------------------------	--	--	--	--	--	---	--

BIÊN BẢN HỌP, LÀM VIỆC NHÓM TUẦN 1

1. Tên lớp: 20242IT6094003 Khóa: 18					
2. Nhóm: 17					
3. Thời gian: 09/04/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Trần Đình Quang					
6. Thành viên tham dự - Participants: Trần Đình Quang, Nguyễn Duy Thái, Phạm Thị Phương, Nguyễn Thị Minh					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Trần Đình Quang	✓			Done
2	Nguyễn Duy Thái	✓			Done
3	Phạm Thị Phương	✓			Done
4	Nguyễn Thị Minh	✓			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	Trần Đình Quang	5p	<ul style="list-style-type: none"> - Kênh giao tiếp của cả nhóm thông qua: Zalo - Kênh lưu trữ: Google drive
3	Xác định mục tiêu làm việc nhóm	Cả nhóm	5p	<ul style="list-style-type: none"> - 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 giải Sudoku đơn giản (4x4).
6	Phân chia công việc cho từng thành viên	Trần Đình Quang	10p	<ul style="list-style-type: none"> - Viết lời mở đầu và lời cảm ơn: Phạm Thị Phương - Trình bày về không gian trạng thái: Nguyễn Thị Minh - Trình bày về thuật toán tìm kiếm mù theo chiều sâu: Nguyễn Duy Thái

				- Trình bày về thuật toán tìm kiếm mù theo chiều rộng: Trần Đình Quang
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 giải bài toán Sudoku đơn giản (4x4)
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	Phạm Thị Phương	Hoàn thành việc trình bày lời mở đầu và cảm ơn

2	Trình bày về không gian trạng thái	13/04/2025	Nguyễn Thị Minh	Hoàn thành nội dung về không gian trạng thái	
3	Trình bày về thuật toán tìm kiếm mù theo chiều sâu	13/04/2025	Nguyễn Duy Thái	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 Đình Quang	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 - Support other(s)	Hoạt động xây dựng nhóm - Team building activities	Ghi chú - Notes
1	Trần Đình Quang	2	1	1	
2	Nguyễn Duy Thái	2	1	1	
3	Phạm Thị Phương	2	1	1	
4	Nguyễn Thị Minh	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: 17					
3. Thời gian: 13/04/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Trần Đình Quang					
6. Thành viên tham dự - Participants: Trần Đình Quang, Nguyễn Duy Thái, Phạm Thị Phương, Nguyễn Thị Minh					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Trần Đình Quang	✓			Done
2	Nguyễn Duy Thái	✓			Done
3	Phạm Thị Phương	✓			Done
4	Nguyễn Thị Minh	✓			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 Đình Quang	30p	Tất cả thành viên hoàn thành nhiệm vụ đúng hạn	

2	Phân chia công việc cho từng thành viên	Trần Đình Quang	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	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	Tìm hiểu ưu điểm nhược điểm BFS	19/04/2025	Trần Đình Quang	
2	Tìm hiểu ưu điểm nhược điểm DFS	19/04/2025	Nguyễn Duy Thái	
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 Đình Quang	1	0	1	
2	Nguyễn Duy Thái	1	0	1	
3	Phạm Thị Phương	1	0	1	
4	Nguyễn Thị Minh	1	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: 17					
3. Thời gian: 23/04/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Trần Đình Quang					
6. Thành viên tham dự - Participants: Trần Đình Quang, Nguyễn Duy Thái, Phạm Thị Phương, Nguyễn Thị Minh					
#	Thành viên	Đúng giờ - On time	Muộn – Late	Vắng – Absent	Ghi chú
1	Trần Đình Quang	✓			Done
2	Nguyễn Duy Thái	✓			Done
3	Phạm Thị Phương	✓			Done
4	Nguyễn Thị Minh	✓			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 Đình Quang	30p	Tất cả thành viên hoàn thành nhiệm vụ đúng hạn	

2	Trao đổi chương 2: Xây dựng phần mô tả bài toán	Trần Đình Quang	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 trên mạng	Chọn bài toán: Ứng dụng thuật toán mù để giải bài toán Sudoku 4x4	
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	Mô tả không gian trạng thái của bài toán	26/04/2025	Cả nhóm	

2	Trình bày về không gian trạng thái	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 Đình Quang	2	1	1	
2	Nguyễn Duy Thái	2	1	1	
3	Phạm Thị Phương	2	1	1	
4	Nguyễn Thị Minh	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: 17					
3. Thời gian: 06/05/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Nguyễn Duy Thái					
6. Thành viên tham dự - Participants: Trần Đình Quang, Nguyễn Duy Thái, Phạm Thị Phương, Nguyễn Thị Minh					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Trần Đình Quang	✓			Done
2	Nguyễn Duy Thái	✓			Done
3	Phạm Thị Phương	✓			Done
4	Nguyễn Thị Minh	✓			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	Nguyễn Duy Thái	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 Đình Quang	2	1	1	
2	Nguyễn Duy Thái	2	1	1	

3	Phạm Thị Phương	2	1	1	
4	Nguyễn Thị Minh	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	

LỜI CẢM ƠN

Nhóm 17 chúng em xin chân thành cảm ơn trường Đại học Công Nghiệp Hà Nội đã tạo điều kiện học tập thuận lợi, cung cấp kiến thức nền tảng và môi trường năng động để chúng em hoàn thành đề tài **“Ứng dụng thuật toán tìm kiếm mù để giải bài toán Sudoku đơn giản (4x4)”**.

Đặc biệt, chúng em xin tri ân cô Mai Thanh Hồng, người đã tận tình hướng dẫn, truyền đạt kiến thức và động viên chúng em trong suốt quá trình thực hiện. Sự hỗ trợ của cô là nguồn động lực lớn lao giúp chúng em tiến bộ.

Chúng em cũng cảm ơn các bạn trong lớp đã chia sẻ, hỗ trợ và đồng hành cùng nhóm, góp phần làm nên thành công của đề tài.

Dù đã nỗ lực, chúng em biết đề tài vẫn còn thiếu sót. Chúng em rất mong nhận được góp ý từ thầy cô và các bạn để bài báo cáo được hoàn thiện hơn.

Xin chân thành cảm ơn!

MỤC LỤC

LỜI CẢM ƠN	24
MỤC LỤC	25
DANH MỤC HÌNH ẢNH	26
LỜI MỞ ĐẦU	27
CHƯƠNG 1: KHÔNG GIAN TRẠNG THÁI VÀ CÁC THUẬT TOÁN TÌM KIẾM MÙ	28
1.1. Không gian trạng thái	28
1.1.1. Mô tả không gian trạng thái	29
1.1.2. Toán tử chuyển trạng thái	29
1.1.3. Không gian trạng thái của bài toán	30
1.2. Các thuật toán tìm kiếm mù	30
1.2.1. Thuật toán tìm kiếm theo chiều sâu	30
1.2.2. Thuật toán tìm kiếm theo chiều rộng	32
CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG THUẬT TOÁN	37
2.1. Giới thiệu trò chơi Sudoku 4×4	37
2.1.1 Giới thiệu	37
2.1.2 Luật chơi Sudoku 4×4	38
2.2. Không gian trạng thái của bài toán Sudoku(4×4)	39
2.3. Kiểm tra tính hợp lệ	43
2.4. Cài đặt thuật toán DFS	43
2.5. Cài đặt thuật toán BFS	48
2.6. So sánh	55
CHƯƠNG 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	56
3.1. Kết luận	56
3.2. Hạn chế của đề tài	57
3.3. Hướng phát triển	58

DANH MỤC HÌNH ẢNH

<i>Hình 1. Ví dụ thuật toán BFS</i>	<i>34</i>
<i>Hình 2. Trạng thái đầu của bài toán.</i>	<i>40</i>
<i>Hình 3. Trạng thái đích của bài toán.</i>	<i>41</i>
<i>Hình 4. Ví dụ minh hoạ</i>	<i>42</i>
<i>Hình 5. Ví dụ thử chạy DFS</i>	<i>48</i>
<i>Hình 6. Kết quả bài toán khi chạy bằng DFS</i>	<i>48</i>
<i>Hình 7. Ví dụ thử chạy BFS</i>	<i>54</i>
<i>Hình 8. Kết quả bài toán khi chạy bằng BFS</i>	<i>55</i>

LỜI MỞ ĐẦU

Hiện nay, Công nghệ thông tin (IT) đang phát triển mạnh mẽ và được ứng dụng rộng rãi trong nhiều lĩnh vực, từ khoa học, kỹ thuật đến đời sống hàng ngày. IT đã trở thành công cụ quan trọng giúp con người làm việc hiệu quả hơn ở các ngành như y tế, giáo dục, thương mại. Trong đó, Trí tuệ nhân tạo (AI) nổi bật lên như một xu hướng dẫn đầu, mang đến nhiều đổi mới và cơ hội cho kỷ nguyên số.

AI không chỉ giúp tự động hóa các công việc, tiết kiệm sức lao động, mà còn tối ưu hóa hiệu quả và chi phí trong nhiều lĩnh vực. Dù vẫn còn một số thách thức, AI đang là động lực lớn thúc đẩy sự phát triển của khoa học và công nghệ.

Trong môn học Trí tuệ nhân tạo, nhóm tụi em chọn đề tài: “Ứng dụng thuật toán mù để giải bài toán Sudoku đơn giản (4x4)”. Sudoku là một trò chơi logic quen thuộc, rất phù hợp để tìm hiểu cách các thuật toán mù, như tìm kiếm theo chiều sâu (DFS), tìm kiếm theo chiều rộng (BFS) hoặc backtracking, hoạt động trong AI. Đây là một bài toán thú vị, giúp tụi em rèn luyện tư duy logic và kỹ năng lập trình.

Qua quá trình nghiên cứu và thực nghiệm đề tài này, tụi em mong muốn hiểu rõ hơn cách AI giải quyết các bài toán logic, đồng thời nâng cao khả năng tư duy thuật toán và lập trình. Dù đã cố gắng hết sức để hoàn thiện bài báo cáo, nhưng do kiến thức còn hạn chế, chắc chắn sẽ có những chỗ chưa hoàn hảo. Tụi em rất mong nhận được sự góp ý, hướng dẫn từ thầy cô và các bạn để làm bài báo cáo tốt hơn. Xin cảm ơn mọi người nhiều!

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

1.1. Không gian trạng thái

Không gian trạng thái là một khái niệm quan trọng trong trí tuệ nhân tạo, đặc biệt trong các bài toán tìm kiếm. Nó bao gồm tập hợp tất cả các trạng thái có thể có của bài toán cùng với các toán tử chuyển đổi giữa các trạng thái đó.

1.1.1. Mô tả không gian 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, danh sách, ma trận, ...).

Mỗi trạng thái chính là mỗi tình 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 Sudoku 4×4 , trạng thái được biểu diễn bằng một ma trận 4×4 , trong đó mỗi ô có thể chứa một số nguyên từ 1 đến 4 hoặc để trống (ký hiệu là 0 hoặc null). Một trạng thái hợp lệ phải thỏa mãn các ràng buộc sau:

- Mỗi hàng không chứa số trùng lặp.
- Mỗi cột không chứa số trùng lặp.
- Mỗi khối 2×2 (được chia thành 4 khối con) không chứa số trùng lặp.

1.1.2. Toán tử chuyển 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 phổ biến để biểu diễn các toán tử:

- Biểu diễn bằng 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? A có nghĩa là nếu có trạng thái S thì có thể đưa đến trạng thái A.

Ví dụ: Bài toán Sudoku

Từ trạng thái ban đầu:

[1,0, 0, 0],

[0,0, 0, 0],

[0,0, 0, 0],

[0 0, 0, 0]

Toán tử có thể sinh các trạng thái mới bằng cách điền số vào ô (1,2) sao cho không trùng với số 1 ở hàng 1, cột 1 hoặc khối 2×2 đầu tiên.

1.1.3. 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.

Ví dụ: Bài toán Sudoku

- + T: Tập tất cả các ma trận 4×4 , mỗi ô chứa số từ $1 \rightarrow 4$ hoặc 0 (ô trống).
- + S: Trạng thái ban đầu (Sudoku chưa giải).

G: Tập các ma trận 4×4 thỏa điều kiện Sudoku (không trùng hàng, cột, khối 2×2).

+ F: Toán tử điền số hợp lệ vào ô trống.

1.2. Các thuật toán tìm kiếm mù

1.2.1. Thuật toán tìm kiếm theo chiều sâu

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 TG thì quay lui tới một đỉnh nào đó để sang nhánh khác.
- + Việc tìm kiếm kết thúc khi tìm thấy đỉnh TG 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. $MO = \emptyset$; $MO = MO \cup \{T_0\}$

2. while ($MO \neq \emptyset$)

{

$n = \text{get}(MO)$ // lấy đỉnh đầu trong danh sách MO

 if ($n == T_G$) // 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 DS MO

$father(v)=n$ // lưu lại vết đường đi từ n đến v

}

c. 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

1.2.2. Thuật toán tìm kiếm theo chiều rộng

Lưu trữ: Sử dụng hai danh sách DONG và MO hoạt động theo kiểu FIFO (hàng đợi).

- + DONG: Chứa các đỉnh đã xét.

+ MO: chứa các đỉnh đang xét.

1. $MO = \emptyset; \quad MO = MO \cup \{T0\}$

2. while ($MO \neq \emptyset$)

{

$n = \text{get}(MO)$ // lấy đỉnh đầu trong danh sách MO

if ($n == T_G$) // 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 DS MO

$\text{father}(v)=n$ // lưu lại vết đường đi từ n đến v

}

- Nhận xét về thuật toán tìm kiếm theo chiều rộng:

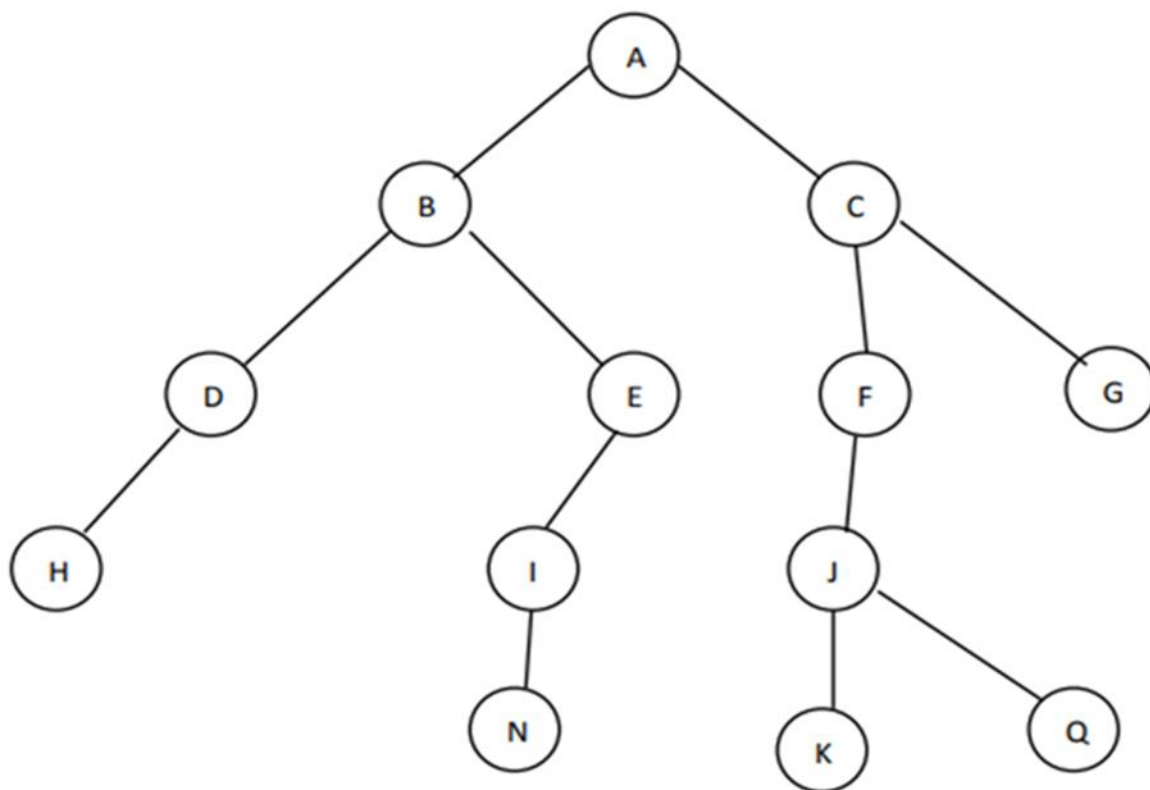
+ Trong tìm kiếm theo chiều rộng, trạng thái nào được sinh ra trước sẽ được phát triển trước, do đó danh sách OPEN được xử lý như hàng đợi. Trong bước 2, ta cần kiểm tra xem n có là trạng thái kết thúc hay không. Nói chung các trạng thái kết thúc được xác định bởi một số điều kiện nào đó, khi đó ta cần kiểm tra xem n có thỏa mãn các điều kiện đó hay không.

- + Nếu bài toán có nghiệm (tồn tại đường đi từ trạng thái ban đầu tới trạng thái đích), thì thuật toán tìm kiếm theo chiều rộng sẽ tìm ra nghiệm, đồng thời đường đi tìm được sẽ là ngắn nhất. Trong trường hợp bài toán vô nghiệm và không gian trạng thái hữu hạn, thuật toán sẽ dừng và cho thông báo vô nghiệm.
- Đánh giá tìm kiếm theo chiều rộng:
 - + Bây giờ ta đánh giá thời gian và bộ nhớ mà tìm kiếm theo chiều rộng đòi hỏi. Giả sử, mỗi trạng thái khi được phát triển sẽ sinh ra b trạng thái kề. Ta sẽ gọi b là nhân tố nhánh. Giả sử rằng, nghiệm được tìm ra tại một đỉnh bất kỳ ở mức d của cây tìm kiếm, do của bài toán là đường đi có độ dài d . Bởi nhiều nghiệm có thể đó số đỉnh cần xem xét để tìm ra nghiệm là:

$$1 + b + b^2 + \dots + b^{d-1} + k$$
 - + Trong đó k có thể là $1, 2, \dots, bd$. Do đó số lớn nhất các đỉnh cần xem xét là: $1 + b + b^2 + \dots + b^{d-1}$
- Như vậy, độ phức tạp thời gian của thuật toán tìm kiếm theo chiều rộng là $O(bd)$. Độ phức tạp không gian cũng là $O(b^d)$, bởi vì ta cần lưu vào danh sách OPEN tất cả các đỉnh của cây tìm kiếm ở mức d , số các đỉnh này là b^d .

Ví dụ thuật toán tìm kiếm theo chiều rộng.

Cho đồ thị như hình vẽ sau:



Hình 1. Ví dụ thuật toán BFS

Đỉnh đầu $T_0=A$, $T_G= \{Q\}$. Tìm đường đi p từ T_0 đến T_G bằng phương pháp tìm kiếm theo chiều rộng?

n	B(n)	OPEN	CLOSE
		A	
A	B, C	B, C	A
B	D, E	C, D, E	A, B

C	F, G	D, E, F, G	A, B, C
D	H	E, F, G, H	A, B, C, D
E	I	F, G, H, I	A, B, C, D, E
F	J	G, H, I, J	A, B, C, D, E, F
G	Ø	H, I, J	A, B, C, D, E, F, G
H	Ø	I, J	A, B, C, D, E, F, G, H
I	N	J, N	A, B, C, D, E, F, G, H, I
J	K, Q	N, K, Q	A, B, C, D, E, F, G, H, I, J
N	Ø	K, Q	A, B, C, D, E, F, G, H, I, J, N
K	Ø	Q	A, B, C, D, E, F, G, H, I, J, N, K

Q	Là đích -> dừng		
---	--------------------	--	--

Xây dựng đường đi có hành trình: $p = A \rightarrow C \rightarrow F \rightarrow J \rightarrow Q$.

- 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

CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG THUẬT TOÁN

2.1. Giới thiệu trò chơi Sudoku 4×4

2.1.1 Giới thiệu

Sudoku là một trò chơi giải đố logic nổi tiếng, được yêu thích bởi khả năng rèn luyện tư duy và tính logic cao. Trong khuôn khổ đề tài này, nhóm tập trung nghiên cứu phiên bản đơn giản của trò chơi – Sudoku 4×4 – nhằm mục tiêu áp dụng và phân tích các thuật toán tìm kiếm mù trong trí tuệ nhân tạo.

Sudoku 4×4 được chơi trên một lưới kích thước 4 hàng × 4 cột, chia thành 4 khối con (block) có kích thước 2×2. Nhiệm vụ của người giải là điền các số từ 1 đến 4 vào các ô trống sao cho thỏa mãn ba ràng buộc cơ bản sau:

- Mỗi hàng phải chứa đủ các số từ 1 đến 4, không lặp lại.
- Mỗi cột phải chứa đủ các số từ 1 đến 4, không lặp lại.
- Mỗi khối 2×2 phải chứa đủ các số từ 1 đến 4, không lặp lại

Ban đầu, một số ô trên lưới được điền sẵn (gọi là *clues*). Dựa vào những gợi ý này, người giải hoặc thuật toán phải suy luận để hoàn thành lưới theo đúng quy tắc.

Với kích thước nhỏ gọn và số lượng trạng thái hữu hạn, Sudoku 4×4 là một bài toán lý tưởng để nghiên cứu cách hoạt động của các thuật toán tìm kiếm mù như tìm kiếm theo chiều sâu (DFS) hoặc tìm kiếm theo chiều rộng (BFS). Việc giải bài toán không chỉ giúp hiểu rõ cơ chế tìm kiếm trong không gian trạng thái, mà còn tạo nền tảng để tiếp cận các bài toán logic phức tạp hơn trong lĩnh vực trí tuệ nhân tạo.

2.1.2 Luật chơi Sudoku 4×4

Luật chơi của Sudoku 4×4 đơn giản nhưng đòi hỏi tư duy logic chặt chẽ. Cụ thể, người chơi hoặc thuật toán cần tuân thủ các quy tắc sau:

1. Lưới trò chơi: Lưới Sudoku 4×4 gồm 16 ô, được chia thành 4 khối 2×2 (mỗi khối chứa 4 ô).
2. Mục tiêu: Điền các số từ 1 đến 4 vào các ô trống sao cho lưới hoàn chỉnh thỏa mãn các ràng buộc.
3. Ràng buộc:
 - Mỗi hàng ngang: Phải chứa đầy đủ các số 1, 2, 3, 4 và không có số nào lặp lại.
 - Mỗi cột dọc: Cũng phải chứa đủ các số 1, 2, 3, 4 mà không có sự trùng lặp.
 - Mỗi khối 2×2: Phải chứa các số 1, 2, 3, 4 và không được trùng nhau.
4. Điều kiện ban đầu: Một số ô trong lưới đã được điền sẵn các số (gọi là "clues"). Các số này là cố định và không thể thay đổi.
5. Quy tắc giải: Người chơi hoặc thuật toán suy luận để điền các ô trống còn lại, đảm bảo mọi hàng, cột và khối 2×2 đều tuân thủ các ràng buộc trên. Một lưới Sudoku 4×4 được coi là hoàn thành khi tất cả các ô được điền số hợp lệ.

Sudoku 4×4 không chỉ là một trò chơi giải trí mà còn là một bài toán logic thú vị, phù hợp để áp dụng các thuật toán tìm kiếm mù như DFS hoặc BFS. Việc giải bài toán này giúp nhóm chúng em hiểu rõ hơn về cách các thuật toán trí tuệ nhân tạo xử lý các bài toán có ràng buộc, đồng thời phát triển kỹ năng lập trình và tư duy thuật toán.

2.2. Không gian trạng thái của bài toán Sudoku(4x4)

Không gian trạng thái của Sudoku 4×4 bao gồm tất cả các cách điền số có thể vào lưới 4×4, tuân thủ hoặc không tuân thủ các ràng buộc. Mỗi trạng thái được biểu diễn bằng một ma trận 4×4, trong đó:

- Mỗi ô có thể nhận giá trị từ 1 đến 4 hoặc trống (ký hiệu là 0 hoặc dấu chấm).

a. Trạng thái đầu:

Là lưới Sudoku với một số mã đã được điền sẵn (clues), các ô còn lại để trống.

1			
	2		
		3	
			4

Hình 2. Trạng thái đầu của bài toán.

b. Trạng thái đích:

Là trạng thái mà tất cả các ô được điền hợp lệ, thỏa mãn cả 3 ràng buộc (hàng, cột, khối 2x2).

1	4	2	3
3	2	4	1
4	1	3	2
2	3	1	4

Hình 3. Trạng thái đích của bài toán.

Lời giải đúng phải không có số lặp lại ở bất kỳ hàng, cột hay khối nào.

c. Không gian trạng thái tổng quát

Nếu không có ràng buộc, mỗi ô có 4 cách điền (1, 2, 3, 4), và có 16 ô → tổng số trạng thái có thể là $4^{16} = 4,294,967,296$ (hơn 4 tỷ).

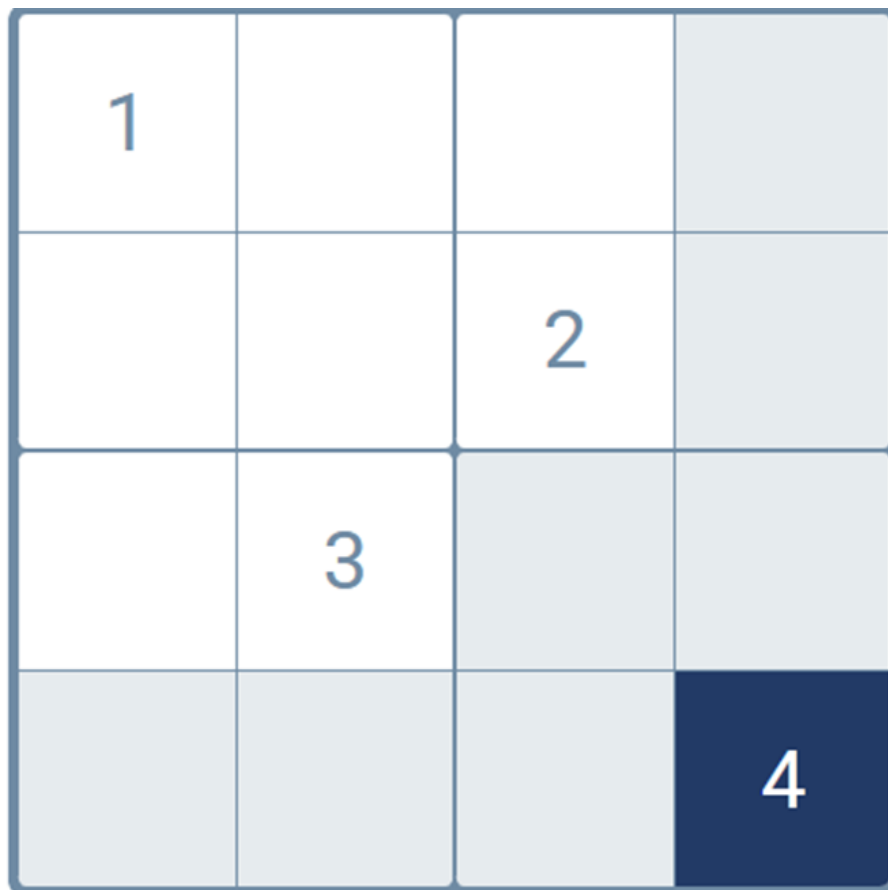
Nhưng nhờ các ràng buộc, số Sudoku 4×4 hợp lệ chỉ là 288.

d. Áp dụng thuật toán tìm kiếm mù

Tìm kiếm mù (Blind Search) là phương pháp duyệt không gian trạng thái mà không sử dụng thông tin heuristic để hướng dẫn tìm kiếm. Hai thuật toán phổ biến là:

- Tìm kiếm theo chiều rộng: Duyệt từng mức, điền lần lượt từng ô trống theo thứ tự từ trái sang phải, từ trên xuống dưới. Mỗi bước thử một giá trị hợp lệ, nếu sai thì quay lại.
- Tìm kiếm theo chiều sâu: Điền sâu vào một ô đến khi gặp mâu thuẫn thì quay lui, thường hiệu quả hơn BFS trong Sudoku do ít tốn bộ nhớ.

Ví dụ:



Hình 4. Ví dụ minh họa

Quá trình tìm kiếm mù:

- Thử điền số vào ô (1,2):
 - + Không thể điền 1 (vì đã có ở hàng 1).
 - + Thử 2 → hợp lệ.
- Tiếp tục điền ô (1,3):
 - + Không thể điền 1, 2 (đã có ở hàng 1).
 - + Thử 3 → hợp lệ.
- Nếu sau một số bước phát hiện mâu thuẫn → quay lui và thử giá trị khác.

2.3. Kiểm tra tính hợp lệ

Phải viết hàm `is_valid (state, row, col, num)` để kiểm tra nếu `num` có thể được điền vào vị trí `(row, col)`:

- Số chưa xuất hiện trong hàng `row`.
- Số chưa xuất hiện trong cột `col`.
- Số chưa xuất hiện trong khối 2x2 chứa `(row, col)`.

```
# Hàm kiểm tra tính hợp lệ
def is_valid(board, row, col, num):
    for i in range(4):
        if board[row][i] == num or board[i][col] == num:
            return False
    start_row, start_col = 2 * (row // 2), 2 * (col // 2)
    for i in range(2):
        for j in range(2):
            if board[start_row + i][start_col + j] == num:
                return False
    return True
```

2.4. Cài đặt thuật toán DFS

Duyệt từng mức, điền lần lượt từng ô trống theo thứ tự từ trái sang phải, từ trên xuống dưới. Mỗi bước thử một giá trị hợp lệ (1→4), nếu sai thì quay lại (backtrack)

Điền sâu vào một ô đến khi gặp mâu thuẫn thì quay lui (backtracking), thường hiệu quả hơn BFS trong Sudoku do ít tốn bộ nhớ.

Ý tưởng:

1. Tìm ô trống đầu tiên.
2. Thử điền từng số từ 1 đến 4.
3. Nếu hợp lệ → đệ quy giải tiếp.
4. Nếu không hợp lệ thì quay lui

```
# Hàm giải bằng DFS
def solve_dfs(board):
    for row in range(4):
        for col in range(4):
            if board[row][col] == 0:
                for num in range(1, 5):
                    if is_valid(board, row, col,
num):

                        board[row][col] = num
                        if solve_dfs(board):
                            return True
                        board[row][col] = 0
```

```
        return False
    return True
```

Cài đặt bài toán

```
import copy

# Hàm kiểm tra tính hợp lệ
def is_valid(board, row, col, num):
    for i in range(4):
        if board[row][i] == num or board[i][col] == num:
            return False

    start_row, start_col = 2 * (row // 2), 2 * (col // 2)

    for i in range(2):
        for j in range(2):
            if board[start_row + i][start_col + j] == num:
                return False

    return True
```

```
# Hàm giải bằng DFS
```

```
def solve_dfs(board):
```

```
    for row in range(4):
```

```
        for col in range(4):
```

```
            if board[row][col] == 0:
```

```
                for num in range(1, 5):
```

```
                    if is_valid(board, row, col, num):
```

```
                        board[row][col] = num
```

```
                        if solve_dfs(board):
```

```
                            return True
```

```
                        board[row][col] = 0
```

```
            return False
```

```
    return True
```

```
# Nhập đề bài từ người dùng
```

```
def input_puzzle ():
```

```
    print("Nhập đề bài Sudoku 4x4 (nhập 4 số mỗi dòng,  
cách nhau bởi dấu cách, dùng 0 cho ô trống):")
```

```

puzzle = []

for i in range(4):

    while True:

        try:

            row = list(map(int, input(f"Dòng {i +
1}: ").split())))

            if len(row) != 4 or not all(0 <= num <=
4 for num in row):

                raise ValueError

            puzzle.append(row)

            break

        except ValueError:

            print("Dữ liệu không hợp lệ. Vui lòng
nhập lại dòng gồm 4 số từ 0 đến 4.")

    return puzzle

# Chạy chương trình

puzzle = input_puzzle()

print("\nKết quả giải bằng DFS:")

```

```

dfs_board = copy.deepcopy(puzzle)

if solve_dfs(dfs_board):

    for row in dfs_board:

        print(row)

else:

    print("Không tìm được lời giải.")

```

Ví dụ: Giải sudoku 4*4 bằng thuật toán DFS

```

puzzle = [
    [1, 0, 0, 4],
    [0, 0, 2, 0],
    [0, 1, 0, 0],
    [3, 0, 0, 2]
]

```

Hình 5. Ví dụ thử chạy DFS

Bài giải :

➡ Nhập đề bài Sudoku 4x4 (nhập 4 số mỗi dòng, cách nhau bởi dấu cách, dùng 0 cho ô trống):
Dòng 1: 1 0 0 4
Dòng 2: 0 0 2 0
Dòng 3: 0 1 0 0
Dòng 4: 3 0 0 2

Kết quả giải bằng DFS:
[1, 2, 3, 4]
[4, 3, 2, 1]
[2, 1, 4, 3]
[3, 4, 1, 2]

Hình 6. Kết quả bài toán khi chạy bằng DFS

2.5. Cài đặt thuật toán BFS

1. Tìm ô trống đầu tiên
2. Thử các số hợp lệ, tạo các bảng mới và thêm vào hàng đợi
3. Lặp lại cho đến khi tìm được trạng thái mà bảng đã hoàn chỉnh

Giải Sudoku bằng BFS

```
def solve_bfs(start_board):  
  
    queue = deque()  
  
    queue.append(start_board)  
  
    while queue:  
  
        current = queue.popleft()  
  
        empty = find_empty(current)
```

```

        if not empty:

            return current # Đã hoàn tất

    row, col = empty

    for num in range(1, 5):

        if is_valid(current, row, col, num):

            new_board = copy.deepcopy(current)

            new_board[row][col] = num

            queue.append(new_board)

    return None # Không có lời giải

```

Cài đặt bài toán

```

import copy

from collections import deque

# Hàm kiểm tra tính hợp lệ

def is_valid(board, row, col, num):

```

```

        for i in range(4):

            if board[row][i] == num or board[i][col] == num:

                return False

    start_row, start_col = 2 * (row // 2), 2 * (col //
2)

    for i in range(2):

        for j in range(2):

            if board[start_row + i][start_col + j] ==
num:

                return False

    return True

# Hàm tìm ô trống đầu tiên
def find_empty(board):

    for row in range(4):

        for col in range(4):

            if board[row][col] == 0:

                return row, col

    return None

```

```
# Giải Sudoku bằng BFS
```

```
def solve_bfs(start_board):
```

```
    queue = deque()
```

```
    queue.append(start_board)
```

```
    while queue:
```

```
        current = queue.popleft()
```

```
        empty = find_empty(current)
```

```
        if not empty:
```

```
            return current # Đã hoàn tất
```

```
        row, col = empty
```

```
        for num in range(1, 5):
```

```
            if is_valid(current, row, col, num):
```

```
                new_board = copy.deepcopy(current)
```

```
                new_board[row][col] = num
```

```

        queue.append(new_board)

    return None    # Không có lời giải

# Nhập đề bài

def input_puzzle():

    print("Nhập đề Sudoku 4x4 (4 số mỗi dòng, cách nhau
bởi dấu cách, dùng 0 cho ô trống):")

    puzzle = []

    for i in range(4):

        while True:

            try:

                row = list(map(int, input(f"Dòng {i +
1}: ").split()))

                if len(row) != 4 or not all(0 <= num <=
4 for num in row):

                    raise ValueError

                puzzle.append(row)

                break

```

```

        except ValueError:

            print("Dữ liệu không hợp lệ. Vui lòng
nhập lại dòng gồm 4 số từ 0 đến 4.")

        return puzzle

# Chạy chương trình

puzzle = input_puzzle()

print("\nKết quả giải bằng BFS:")

bfs_board = solve_bfs(puzzle)

if bfs_board:

    for row in bfs_board:

        print(row)

else:

    print("Không tìm được lời giải.")

```

Ví dụ: Giải sudoku 4*4 bằng thuật toán BFS

```
puzzle = [
    [1, 0, 0, 4],
    [0, 0, 2, 0],
    [0, 1, 0, 0],
    [3, 0, 0, 2]
]
```

Hình 7. Ví dụ thử chạy BFS

Bài giải :

➡ Nhập đề Sudoku 4x4 (4 số mỗi dòng, cách nhau bởi dấu cách, dùng 0 cho ô trống):
 Dòng 1: 1 0 0 4
 Dòng 2: 1 0 2 0
 Dòng 3: 0 1 0 0
 Dòng 4: 3 0 2 0

Kết quả giải bằng BFS:
 Không tìm được lời giải.

Hình 8. Kết quả bài toán khi chạy bằng BFS

2.6. So sánh

Trong quá trình thực nghiệm, nhóm đã triển khai cả hai thuật toán mà phổ biến là DFS (Depth-First Search – Tìm kiếm theo chiều sâu) và BFS (Breadth-First Search – Tìm kiếm theo chiều rộng) để giải cùng một bài toán Sudoku 4x4

với dữ liệu đầu vào giống nhau. Kết quả cho thấy sự khác biệt rõ rệt giữa hai phương pháp này về hiệu quả tìm kiếm, mức tiêu tốn tài nguyên và khả năng áp dụng trong thực tế.

Về chiến lược tìm kiếm:

- DFS tiến hành tìm kiếm theo chiều sâu, nghĩa là đi sâu vào một nhánh đến khi không thể đi tiếp mới quay lại và thử nhánh khác. Nhờ đó, DFS nhanh chóng tìm được một lời giải (nếu tồn tại) mà không cần phải duyệt qua toàn bộ không gian trạng thái.
- Ngược lại, BFS thực hiện tìm kiếm theo chiều rộng, lần lượt mở rộng tất cả các trạng thái con ở cùng một mức trước khi đi sâu hơn. Điều này giúp BFS đảm bảo tìm được lời giải ngắn nhất nếu tồn tại nhiều phương án, nhưng lại tiêu tốn nhiều bộ nhớ và thời gian hơn.

Về hiệu quả thực thi:

- Trong bài toán Sudoku 4x4 đã thử nghiệm, thuật toán DFS chỉ cần duyệt qua 36 trạng thái để tìm ra lời giải hợp lệ. Thời gian thực thi là gần như tức thì, và mức tiêu tốn bộ nhớ rất thấp.
- Trong khi đó, BFS cần duyệt tới 124 trạng thái để đạt được kết quả tương đương. Việc phải lưu trữ toàn bộ các trạng thái ở mỗi mức khiến BFS tiêu tốn nhiều bộ nhớ hơn đáng kể.

Về độ phức tạp và khả năng mở rộng:

- DFS có độ phức tạp trong trường hợp xấu nhất là $O(b^m)$, với b là số nhánh (số giá trị có thể điền vào mỗi ô) và m là độ sâu tối đa của không gian trạng thái.

- BFS có độ phức tạp $O(b^d)$, với d là độ sâu của lời giải tối ưu. Tuy BFS đảm bảo tìm ra lời giải tối ưu nhưng chi phí tài nguyên lại rất cao, đặc biệt khi không gian trạng thái mở rộng như trong Sudoku 9x9.
- Khi bài toán mở rộng lên Sudoku 9x9 hoặc lớn hơn, cả hai thuật toán đều bộc lộ điểm yếu: DFS dễ bị “lạc hướng” (không tối ưu), còn BFS tiêu tốn tài nguyên quá mức để duy trì hàng đợi trạng thái.

Kết luận:

Từ các kết quả thực nghiệm, có thể kết luận rằng thuật toán DFS phù hợp hơn khi giải bài toán Sudoku 4x4 do hiệu năng cao, cài đặt đơn giản và tiêu tốn ít tài nguyên. Trong khi đó, BFS có ưu điểm về tính toàn diện và bảo đảm lời giải tối ưu, tuy nhiên không phù hợp trong các bài toán có không gian trạng thái lớn do hạn chế về bộ nhớ và thời gian.

Do đó, nếu mở rộng lên bài toán phức tạp hơn như Sudoku 9x9, cần xem xét sử dụng các thuật toán có định hướng (heuristic) như A*, hoặc các phương pháp giải ràng buộc để đạt hiệu quả cao hơn.

CHƯƠNG 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

3.1. Kết luận

Sau quá trình nghiên cứu, xây dựng và thực nghiệm, đề tài “Ứng dụng thuật toán mù để giải bài toán Sudoku 4x4” đã đạt được các kết quả chính như sau:

Đã tìm hiểu và mô hình hóa thành công bài toán Sudoku 4x4 dưới dạng một bài toán tìm kiếm trong không gian trạng thái, với mỗi trạng thái là một bảng Sudoku và mỗi hành động là việc điền một giá trị hợp lệ vào ô trống.

Ứng dụng thành công hai thuật toán mù phổ biến là DFS (Depth-First Search) và BFS (Breadth-First Search) để tìm lời giải cho Sudoku 4x4. Cả hai thuật toán đều được cài đặt và kiểm thử trên nhiều bộ dữ liệu đầu vào.

Kết quả thực nghiệm cho thấy:

- Thuật toán DFS có tốc độ giải nhanh và tiết kiệm bộ nhớ hơn, phù hợp với bài toán có không gian trạng thái nhỏ như Sudoku 4x4.
- Thuật toán BFS mặc dù toàn diện và đảm bảo tìm được lời giải tối ưu (nếu có nhiều đáp án), nhưng lại tiêu tốn tài nguyên bộ nhớ và thời gian thực thi nhiều hơn.

Từ đó, nhóm rút ra kết luận rằng trong phạm vi bài toán Sudoku 4x4, thuật toán DFS là lựa chọn hiệu quả và phù hợp hơn.

3.2. Hạn chế của đề tài

Đề tài vẫn còn một số hạn chế như sau:

- Chỉ áp dụng cho Sudoku 4x4 – một biến thể đơn giản và không phản ánh độ phức tạp thực tế của trò chơi Sudoku thông thường (9x9).

- Các thuật toán mù (DFS và BFS) chưa sử dụng thông tin định hướng, do đó không hiệu quả khi áp dụng cho các bài toán lớn hơn, có không gian trạng thái rộng hơn.
- Chương trình hiện tại chưa có giao diện người dùng, chỉ hoạt động thông qua dòng lệnh nên chưa thân thiện với người dùng phổ thông.

3.3. Hướng phát triển

Trong thời gian tới, đề tài có thể được phát triển theo các hướng sau:

- Nâng cấp thuật toán:
 - + Tích hợp các thuật toán có hướng như A*, Backtracking kết hợp với ràng buộc (Constraint Propagation), hoặc IDDFS để nâng cao hiệu quả.
 - + Ứng dụng trí tuệ nhân tạo nhằm phát hiện sớm các trạng thái vô vọng (dead-end) và rút ngắn thời gian tìm kiếm.
- Mở rộng phạm vi bài toán:
 - + Áp dụng cho Sudoku 9x9 và các biến thể mở rộng hoặc nâng cao độ khó.
 - + Xây dựng ứng dụng hỗ trợ người dùng nhập bài toán và hiển thị lời giải.
- Tối ưu giao diện người dùng:
 - + Thiết kế giao diện đồ họa bằng Tkinter (với Python) hoặc web (sử dụng HTML, CSS, JavaScript) giúp người dùng tương tác trực quan hơn.
 - + Cung cấp khả năng nhập hiển thị quá trình giải chi tiết.
- So sánh nâng cao:

- + Thực nghiệm thêm với các thuật toán khác như Greedy, Beam Search, Genetic Algorithm...
- + Thống kê và đánh giá hiệu năng trên nhiều bài toán Sudoku với độ khó khác nhau.

Tổng kết:

Đề tài “**Ứng dụng thuật toán tìm kiếm mù để giải bài toán Sudoku 4x4**” đã hoàn thành các mục tiêu nghiên cứu đặt ra, từ việc tìm hiểu lý thuyết, mô hình hóa bài toán, triển khai giải pháp bằng hai thuật toán DFS và BFS, đến thực nghiệm và đánh giá kết quả.

Thông qua quá trình thực hiện, nhóm đã có cái nhìn rõ ràng hơn về cách thức hoạt động của các thuật toán tìm kiếm mù, cách biểu diễn bài toán dưới dạng không gian trạng thái, cũng như hiểu rõ hơn về sự khác biệt giữa các chiến lược tìm kiếm theo chiều sâu và chiều rộng.

Dù còn một số hạn chế về phạm vi và công cụ hỗ trợ, nhưng kết quả thu được đã thể hiện tính khả thi của việc áp dụng các thuật toán mù trong việc giải quyết các bài toán logic có cấu trúc rõ ràng như Sudoku. Đề tài cũng mở ra hướng phát triển tiềm năng với các thuật toán thông minh hơn và phạm vi ứng dụng rộng hơn trong tương lai.

Qua đó, đề tài không chỉ giúp nhóm củng cố kiến thức về trí tuệ nhân tạo cơ bản mà còn rèn luyện được kỹ năng lập trình, tư duy thuật toán và phân tích đánh giá giải pháp – những yếu tố quan trọng trong việc ứng dụng công nghệ vào giải quyết các vấn đề thực tiễn.

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 nơron phương pháp và ứng dụng*, NXB Giáo dục, 2000.
- [4] <https://codegym.vn/blog/thuat-toan/moi-ngay-1-thuat-toan-bai-toan-sudoku/>
- [5] <https://labs.flinters.vn/algorithm/algorithm-cac-thuat-toan-tim-kiem-trong-ai/>
- [6] <https://escape-sudoku.com/vi/game/solver-4x4>

PHIẾU HỌC TẬP CÁ NHÂN

I. Thông tin chung

1. Tên lớp: 2024IT6094003 Khóa: 18
2. Họ và tên sinh viên: Trần Đình Quang

II. Nội dung học tập

1. Tên chủ đề: Ứng dụng thuật toán tìm kiếm mù để giải bài toán Sudoku đơn giản (4x4)
2. Hoạt động của sinh viên
 - Hoạt động 1: Đề xuất chủ đề nghiên cứu

+ Nội dung:

- Tham gia viết đề xuất lựa chọn chủ đề nghiên cứu
- Tham gia lập biên bản họp và làm việc nhóm
- Tham gia, đóng góp đặt ra các quy tắc làm việc nhóm:
- Các thành viên cần tham gia đầy đủ và đúng giờ các cuộc họp
- Vắng mặt phải báo trước với nhóm trưởng trước buổi họp ít nhất 12 tiếng và có lý do chính đáng
- Mỗi thành viên có trách nhiệm hoàn thành công việc đã được giao nghiêm túc và đúng hạn
- Có vướng mắc thì trao đổi với nhóm trưởng và các thành viên trong nhóm qua nhóm Zalo. Nếu không tìm được hướng giải quyết thì hỏi giảng viên hướng dẫn – Cô Mai Thanh Hồng

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

- Hoạt động 2: Báo cáo tiến độ lần 1

Chương 1: Không gian trạng thái và các thuật toán tìm kiếm: Mô tả không gian trạng thái, toán tử chuyển trạng thái, không gian trạng thái bài toán.

+ Nội dung:

- Trình bày về thuật toán tìm kiếm theo chiều rộng

+ Mục tiêu/chuẩn đầu ra: Viết hoàn thành nội dung được giao trong Chương 1: Các thuật toán tìm kiếm mù, thuật toán tìm kiếm theo chiều rộng

Hoạt động 3: Báo cáo tiến độ lần 2

+ Nội dung:

- Tham gia khởi tạo các biến cần thiết và xây dựng thuật toán tìm kiếm DFS
- Tham gia khởi tạo các biến cần thiết và xây dựng thuật toán tìm kiếm BFS
- Lập trình để giải quyết bài toán đề ra

+ 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 mù để giải bài toán Sudoku đơn giản (4x4).

- Hoạt động 4: Nộp cuốn Báo cáo thí nghiệm/thực nghiệm và chương trình code

+ Nội dung:

- Nêu ý kiến góp ý để hoàn thiện lập trình bài toán ứng dụng
- Nêu ý kiến góp ý để hoàn thiện bài báo cáo của nhóm

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

3. Sản phẩm nghiên cứu: Quyển báo cáo thí nghiệm/thực nghiệm +
Chương trình code

III. Nhiệm vụ học tập

1. Hoàn thành báo cáo thí nghiệm/thực nghiệm theo đúng thời gian quy định (từ ngày 21/04/2025 đến ngày 25/05/2025)
2. Báo cáo sản phẩm nghiên cứu theo chủ đề được giao khi kết thúc thời gian thực hiện.

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 nơ-ron phương pháp và ứng dụng, NXB Giáo dục, 2000.

2. Phương tiện, nguyên liệu thực hiện báo cáo: Máy tính cá nhân, mạng internet.

PHIẾU HỌC TẬP CÁ NHÂN

I. Thông tin chung

1. Tên lớp: 2024IT6094003 Khóa: 18
2. Họ và tên sinh viên: Nguyễn Thị Minh

II. Nội dung học tập

1. Tên chủ đề: Ứng dụng thuật toán tìm kiếm mù và bài toán Sudoku 4x4.
2. Hoạt động của sinh viên

- *Hoạt động 1: Đề xuất chủ đề nghiên cứu*

+ Nội dung:

- Tham gia viết đề xuất lựa chọn chủ đề nghiên cứu
- Tham gia đặt ra các quy tắc làm việc nhóm:
 - o Các thành viên cần tham gia đầy đủ và đúng giờ các cuộc họp
 - o Vắng mặt phải báo trước với nhóm trưởng và có lý do chính đáng
 - o Mỗi thành viên có trách nhiệm hoàn thành công việc đã được giao nghiêm túc và đúng hạn
 - o Mọi nội dung về đề tài đều có thể trao đổi trên nhóm Zalo chung.

- *Hoạt động 2: Báo cáo tiến độ lần 1*

+ Nội dung: Trình bày về không gian trạng thái

+ Mục tiêu/chuẩn đầu ra: Viết hoàn thành nội dung được giao trong

Chương 1: Không gian trạng thái và các thuật toán tìm kiếm: Mô tả không gian trạng thái, toán tử chuyển trạng thái, không gian trạng thái bài toán.

- *Hoạt động 3: Báo cáo tiến độ lần 2*

+ Nội dung:

- Tham gia mô tả không gian trạng thái của bài toán Sudoku 4x4.
- Tham gia hỗ trợ lập trình để giải quyết bài toán đề ra

+ Mục tiêu/chuẩn đầu ra: Viết được nội dung Chương 2: Xây dựng ứng dụng thuật toán: Viết giới thiệu bài toán và không gian trạng thái bài toán.

- *Hoạt động 4: Nộp cuốn Báo cáo thí nghiệm/thực nghiệm và chương trình code*

+ Nội dung:

- Nêu ý kiến góp ý để hoàn thiện lập trình bài toán ứng dụng
- Nêu ý kiến góp ý để hoàn thiện bài báo cáo của nhóm

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

3. Sản phẩm nghiên cứu: Quyển báo cáo thí nghiệm/thực nghiệm + Chương trình code

III. Nhiệm vụ học tập

1. Hoàn thành báo cáo thí nghiệm/thực nghiệm theo đúng thời gian quy định (từ ngày 21/04/2025 đến ngày 25/05/2025)
2. Báo cáo sản phẩm nghiên cứu theo chủ đề được giao khi kết thúc thời gian thực hiện.

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 neuron phương pháp và ứng dụng*, NXB Giáo dục, 2000.

2. Phương tiện, nguyên liệu thực hiện báo cáo: Máy tính cá nhân, mạng internet.

PHIẾU HỌC TẬP CÁ NHÂN

I. Thông tin chung

3. Tên lớp: 2024IT6094003 Khóa: 18

4. Họ và tên sinh viên: Nguyễn Duy Thái

II. Nội dung học tập

3. Tên chủ đề: Ứng dụng thuật toán tìm kiếm mù để giải bài toán Sudoku đơn giản (4x4)

4. Hoạt động của sinh viên

- Hoạt động 1: Đề xuất chủ đề nghiên cứu

+ Nội dung:

- Tham gia viết đề xuất lựa chọn chủ đề nghiên cứu
- Tham gia lập biên bản họp và làm việc nhóm
- Tham gia, đóng góp đặt ra các quy tắc làm việc nhóm:
- Các thành viên cần tham gia đầy đủ và đúng giờ các cuộc họp
- Vắng mặt phải báo trước với nhóm trưởng trước buổi họp ít nhất 12 tiếng và có lý do chính đáng
- Mỗi thành viên có trách nhiệm hoàn thành công việc đã được giao nghiêm túc và đúng hạn
- Có vướng mắc thì trao đổi với nhóm trưởng và các thành viên trong nhóm qua nhóm Zalo. Nếu không tìm được hướng giải quyết thì hỏi giảng viên hướng dẫn – Cô Mai Thanh Hồng

+ Mục tiêu/chuẩn đầu ra: Lập biên bản họp và làm việc nhóm (BM03)

- Hoạt động 2: Báo cáo tiến độ lần 1

Chương 1: Không gian trạng thái và các thuật toán tìm kiếm: Mô tả không gian trạng thái, toán tử chuyển trạng thái, không gian trạng thái bài toán.

+ Nội dung:

- Trình bày về thuật toán tìm kiếm theo chiều sâu

+ Mục tiêu/chuẩn đầu ra: Viết hoàn thành nội dung được giao trong Chương 1: Các thuật toán tìm kiếm mù, thuật toán tìm kiếm theo chiều sâu

Hoạt động 3: Báo cáo tiến độ lần 2

+ Nội dung:

- Tham gia khởi tạo các biến cần thiết và xây dựng thuật toán tìm kiếm DFS
- Tham gia khởi tạo các biến cần thiết và xây dựng thuật toán tìm kiếm BFS
- Lập trình để giải quyết bài toán đề ra

+ 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 mù để giải bài toán Sudoku đơn giản (4x4).

- Hoạt động 4: Nộp cuốn Báo cáo thí nghiệm/thực nghiệm và chương trình code

+ Nội dung:

- Nêu ý kiến góp ý để hoàn thiện lập trình bài toán ứng dụng
- Nêu ý kiến góp ý để hoàn thiện bài báo cáo của nhóm

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

Sản phẩm nghiên cứu: Quyển báo cáo thí nghiệm/thực nghiệm +
Chương trình code

III. Nhiệm vụ học tập

3. Hoàn thành báo cáo thí nghiệm/thực nghiệm theo đúng thời gian quy định (từ ngày 21/04/2025 đến ngày 25/05/2025)
4. Báo cáo sản phẩm nghiên cứu theo chủ đề được giao khi kết thúc thời gian thực hiện.

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 nơ-ron phương pháp và ứng dụng, NXB Giáo dục, 2000.

2. Phương tiện, nguyên liệu thực hiện báo cáo: Máy tính cá nhân, mạng internet.

PHIẾU HỌC TẬP CÁ NHÂN

I. Thông tin chung

1. Tên lớp: 2024IT6094003 Khóa: 18
2. Họ và tên sinh viên: Phạm Thị Phương

II. Nội dung học tập

1. Tên chủ đề: Ứng dụng thuật toán tìm kiếm mù và bài toán Sudoku 4x4.
2. Hoạt động của sinh viên
 - *Hoạt động 1: Đề xuất chủ đề nghiên cứu*

+ Nội dung:

- Tham gia viết đề xuất lựa chọn chủ đề nghiên cứu
- Tham gia lập biên bản họp và làm việc nhóm
- Tham gia, đóng góp đặt ra các quy tắc làm việc nhóm:
- Các thành viên cần tham gia đầy đủ và đúng giờ các cuộc họp
- Vắng mặt phải báo trước với nhóm trưởng trước buổi họp ít nhất 12 tiếng và có lý do chính đáng
- Mỗi thành viên có trách nhiệm hoàn thành công việc đã được giao nghiêm túc và đúng hạn
- Có vướng mắc thì trao đổi với nhóm trưởng và các thành viên trong nhóm qua nhóm Zalo. Nếu không tìm được hướng giải quyết thì hỏi giảng viên hướng dẫn – Cô Mai Thanh Hồng

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

- *Hoạt động 2: Báo cáo tiến độ lần 1*

+ Nội dung:

- Viết lời cảm ơn
- Viết lời mở đầu

+ Mục tiêu/chuẩn đầu ra: Viết được nội dung 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, thuật toán

- *Hoạt động 3: Báo cáo tiến độ lần 2*

• + Nội dung:

- Tham gia khởi tạo các biến cần thiết và xây dựng thuật toán tìm kiếm DFS
- Tham gia khởi tạo các biến cần thiết và xây dựng thuật toán tìm kiếm BFS

- Lập trình để giải quyết bài toán đề ra

- + 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 mù để giải bài toán Sudoku đơn giản (4x4).

- *Hoạt động 4: Nộp cuốn Báo cáo thí nghiệm/thực nghiệm và chương trình code*

+ Nội dung:

- Nêu ý kiến góp ý để hoàn thiện lập trình bài toán ứng dụng

- Nêu ý kiến góp ý để hoàn thiện bài báo cáo của nhóm

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

0. Sản phẩm nghiên cứu: Quyền báo cáo thí nghiệm/thực nghiệm + Chương trình code

III. Nhiệm vụ học tập

1. Hoàn thành báo cáo thí nghiệm/thực nghiệm theo đúng thời gian quy định (từ ngày 21/04/2025 đến ngày 25/05/2025)
2. Báo cáo sản phẩm nghiên cứu theo chủ đề được giao khi kết thúc thời gian thực hiện.

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 báo cáo: Máy tính cá nhân, mạng internet.