

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

===== ★ ★ ★ =====



BÁO CÁO BTL Trí Tuệ Nhân Tạo

Tìm hiểu thuật toán tìm kiếm heuristics và ứng dụng vào bài toán tìm đường đi ngắn nhất.

GVHD: ThS. Mai Thanh Hồng

Nhóm lớp: 6 - 20242IT6094007

Thành viên: Đinh Hải Anh - 2023605264

Nguyễn Đức Trung - 2023600531

Hà Thành - 2023605169

Nguyễn Minh Ngọc - 2023607654

Hà Nội , 2025

BÁO CÁO HỌC TẬP NHÓM

Tên lớp: 20242IT6094007

Khoá: 18

Tên nhóm: 6

Họ và tên thành viên trong nhóm:

1.Họ và tên SV: Nguyễn Đức Trung MãSV: 2023600531

2.Họ và tên SV: Đinh Hải Anh Mã SV: 2023605264

3.Họ và tên SV: Hà Thành Mã SV: 2023605169

4.Họ và tên SV: Nguyễn Minh Ngọc Mã SV: 2023607654

Tên chủ đề: Tìm hiểu thuật toán tìm kiếm Heuristics và ứng dụng vào bài toán tìm đường đi ngắn nhất

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 - Trình bày về không gian trạng thái	Các thành viên hoàn thành đúng thời hạn và đầy đủ nội dung	
2	Cả nhóm	- Trình bày về tổng quan về thuật toán tìm kiếm Heuristic và một số thuật toán tìm kiếm Heuristic	Các thành viên hoàn thành đúng thời hạn và đầy đủ nội dung	

3	Cả nhóm	-Ứng dụng kiến thức đã tìm hiểu về thuật toán để áp dụng vào bài toán tìm đường đi ngắn nhất	Các thành viên hoàn thành đúng thời hạn và đầy đủ nội dung	
4	Cả nhóm	xem và sửa BTL	Các thành viên hoàn thành đúng thời hạn và đầy đủ nội dung	
5	Cả nhóm	xem và sửa BTL	Các thành viên hoàn thành đúng thời hạn và đầy đủ nội dung	
6	Cả nhóm	Xem và sửa BTL	Các thành viên hoàn thành đúng thời hạn và đầy đủ nội dung	
7	Cả nhóm	Hoàn thành BTL	Các thành viên hoàn thành đúng thời hạn và đầy đủ nội dung	

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: 20242IT6094007 Khoá: 18
2. Tên nhóm: 6
3. Họ và tên thành viên trong nhóm:
 1. Họ và tên SV: Nguyễn Đức Trung MãSV: 2023600531
 2. Họ và tên SV: Đinh Hải Anh Mã SV: 2023605264
 3. Họ và tên SV: Hà Thành Mã SV: 2023605169
 4. Họ và tên SV: Nguyễn Minh Ngọc Mã SV: 2023607654

II. Nội dung học tập:

1. Tên chủ đề: Tìm hiểu thuật toán tìm kiếm heuristic và ứng dụng vào bài toán tìm đường đi ngắn nhất

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
- 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 heuristics
- Giới thiệu tổng quan về không gian trạng thái, toán tử chuyển trạng thái, 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 Heuristics và ứng dụng vào bài toán tìm đường đi ngắn nhất

- 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 25 / 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.

BIÊN BẢN HỌP NHÓM

BIÊN BẢN HỌP, LÀM VIỆC NHÓM TUẦN 1					
1. Tên lớp: 20242IT6094007 Khóa: 18					
2. Nhóm: 6					
3. Thời gian: 06/04/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Nguyễn Đức Trung					
6. Thành viên tham dự - Participants: Nguyễn Đức Trung, Đinh Hải Anh , Hà Thành , Nguyễn Minh Ngọc					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Nguyễn Đức Trung	✓			Done
2	Đinh Hải Anh	✓			Done
3	Hà Thành	✓			Done
4	Nguyễn Minh Ngọc	✓			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	- Nhóm trưởng	
2	Thiết lập kênh giao tiếp, lưu trữ	Nguyễn Đức Trung	5p	- 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	- 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	Xác định nội dung cần làm	Cả nhóm	30p	Thống nhất nội dung cần làm
5	Phân chia công việc cho từng thành viên	Nguyễn Đức Trung	15p	Xem và sửa BTL

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 tìm kiếm mù, thuật toán tìm kiếm heuristic	thuật toán heuristic	Đề tài ứng dụng thuật toán tìm kiếm heuristic vào bài toán tìm đường đi ngắn nhất
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 + trình bày về toán tử trạng thái	10/04/2025	Nguyễn Minh Ngọc	Hoàn thiện
2	Trình bày về mô tả trạng thái	10/04/2025	Đinh Hải Anh	Hoàn thiện
3	Trình bày về không gian của trạng thái	10/04/2025	Hà Thành	Hoàn thiện
4	Tổng hợp nội dung và thống nhất bố cục	10/04/2025	Nguyễn Đức Trung	Hoàn thiện

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	Nguyễn Đức Trung	4	1	1	
2	Đinh Hải Anh	1	1	1	
3	Hà Thành	1	1	1	
4	Nguyễn Minh Ngọc	1	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: 20242IT6094007 Khóa: 18					
2. Nhóm: 6					
3. Thời gian: 13/04/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Nguyễn Đức Trung					
6. Thành viên tham dự - Participants: Nguyễn Đức Trung, Đinh Hải Anh , Hà Thành , Nguyễn Minh Ngọc					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Nguyễn Đức Trung	✓			Done
2	Đinh Hải Anh	✓			Done
3	Hà Thành	✓			Done
4	Nguyễn Minh Ngọc	✓			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	- Nhóm trưởng	
2	Thiết lập kênh giao tiếp, lưu trữ	Nguyễn Đức Trung	5p	- 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	- 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	Xác định nội dung cần làm	Cả nhóm	30p	Thống nhất nội dung cần làm
5	Phân chia công việc cho từng thành viên	Nguyễn Đức Trung	15p	Xem và sửa BTL

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 tìm kiếm mù, thuật toán tìm kiếm heuristic	thuật toán heuristic	Đề tài ứng dụng thuật toán tìm kiếm heuristic vào bài toán tìm đường đi ngắn nhất
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	Trình bày về thuật toán BEFS	19/04/2025	Nguyễn Minh Ngọc	Hoàn thiện	
2	Trình bày về thuật toán AKT	19/04/2025	Đinh Hải Anh	Hoàn thiện	
3	Trình bày về thuật toán AT	19/04/2025	Hà Thành	Hoàn thiện	
4	Trình bày về thuật toán A*	19/04/2025	Nguyễn Đức Trung	Hoàn thiện	
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	Nguyễn Đức Trung	4	1	1	
2	Đinh Hải Anh	1	1	1	
3	Hà Thành	1	1	1	
4	Nguyễn Minh Ngọc	1	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 3					
1. Tên lớp: 20242IT6094007 Khóa: 18					
2. Nhóm: 6					
3. Thời gian: 20/04/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Nguyễn Đức Trung					
6. Thành viên tham dự - Participants: Nguyễn Đức Trung, Đinh Hải Anh , Hà Thành , Nguyễn Minh Ngọc					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Nguyễn Đức Trung	✓			Done
2	Đinh Hải Anh	✓			Done
3	Hà Thành	✓			Done
4	Nguyễn Minh Ngọc	✓			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	- Nhóm trưởng	
2	Thiết lập kênh giao tiếp, lưu trữ	Nguyễn Đức Trung	5p	- 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	- 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	Xác định nội dung cần làm	Cả nhóm	30p	Thống nhất nội dung cần làm
5	Phân chia công việc cho từng thành viên	Nguyễn Đức Trung	15p	Xem và sửa BTL

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 tìm kiếm mù, thuật toán tìm kiếm heuristic	thuật toán heuristic	Đề tài ứng dụng thuật toán tìm kiếm heuristic vào bài toán tìm đường đi ngắn nhất
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	Cài đặt thuật toán tìm kiếm đường đi ngắn nhất	26/04/2025	Nguyễn Minh Ngọc	Hoàn thiện
2	Cài đặt thuật toán tìm kiếm đường đi ngắn nhất	26/04/2025	Đinh Hải Anh	Hoàn thiện
3	Cài đặt thuật toán tìm kiếm đường đi ngắn nhất	26/04/2025	Hà Thành	Hoàn thiện
4	Cài đặt thuật toán tìm kiếm đường đi ngắn nhất	26/04/2025	Nguyễn Đức Trung	Hoàn thiện

Đóng góp nhóm - Team contribution

#	Thành viên - Member	Ý tưởng, giải pháp - Ideal	Hỗ trợ người khác - Support other(s)	Hoạt động xây dựng nhóm - Team building activities	Ghi chú - Notes
1	Nguyễn Đức Trung	4	1	1	
2	Đinh Hải Anh	1	1	1	
3	Hà Thành	1	1	1	
4	Nguyễn Minh Ngọc	1	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: 20242IT6094007 Khóa: 18					
2. Nhóm: 6					
3. Thời gian: 27/04/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Nguyễn Đức Trung					
6. Thành viên tham dự - Participants: Nguyễn Đức Trung, Đinh Hải Anh , Hà Thành , Nguyễn Minh Ngọc					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Nguyễn Đức Trung	✓			Done
2	Đinh Hải Anh	✓			Done
3	Hà Thành	✓			Done
4	Nguyễn Minh Ngọc	✓			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	- Nhóm trưởng	
2	Thiết lập kênh giao tiếp, lưu trữ	Nguyễn Đức Trung	5p	- 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	- 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	Xác định nội dung cần làm	Cả nhóm	30p	Thông nhất nội dung cần làm
5	Phân chia công việc cho từng thành viên	Nguyễn Đức Trung	15p	Xem và sửa BTL

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 tìm kiếm mù, thuật toán tìm kiếm heuristic	thuật toán heuristic	Đề tài ứng dụng thuật toán tìm kiếm heuristic vào bài toán tìm đường đi ngắn nhất
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	Cài đặt thuật toán tìm kiếm đường đi ngắn nhất	03/05/2025	Nguyễn Minh Ngọc	Hoàn thiện
2	Cài đặt thuật toán tìm kiếm đường đi ngắn nhất	03/05/2025	Đinh Hải Anh	Hoàn thiện
3	Cài đặt thuật toán tìm kiếm đường đi ngắn nhất	03/05/2025	Hà Thành	Hoàn thiện
4	Cài đặt thuật toán tìm kiếm đường đi ngắn nhất	03/05/2025	Nguyễn Đức Trung	Hoàn thiện

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	Nguyễn Đức Trung	4	1	1	
2	Đinh Hải Anh	1	1	1	
3	Hà Thành	1	1	1	
4	Nguyễn Minh Ngọc	1	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 5					
1. Tên lớp: 20242IT6094007 Khóa: 18					
2. Nhóm: 6					
3. Thời gian: 04/05/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Nguyễn Đức Trung					
6. Thành viên tham dự - Participants: Nguyễn Đức Trung, Đinh Hải Anh , Hà Thành , Nguyễn Minh Ngọc					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Nguyễn Đức Trung	✓			Done
2	Đinh Hải Anh	✓			Done
3	Hà Thành	✓			Done
4	Nguyễn Minh Ngọc	✓			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	- Nhóm trưởng	
2	Thiết lập kênh giao tiếp, lưu trữ	Nguyễn Đức Trung	5p	- 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	- 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	Xác định nội dung cần làm	Cả nhóm	30p	Thống nhất nội dung cần làm
5	Phân chia công việc cho từng thành viên	Nguyễn Đức Trung	15p	Xem và sửa BTL

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 tìm kiếm mù, thuật toán tìm kiếm heuristic	thuật toán heuristic	Đề tài ứng dụng thuật toán tìm kiếm heuristic vào bài toán tìm đường đi ngắn nhất
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	Cài đặt thuật toán của bài toán tìm đường đi ngắn nhất	10/05/2025	Nguyễn Đức Trung	Xem và sửa BTL	
2	Cài đặt thuật toán của bài toán tìm đường đi ngắn nhất	10/05/2025	Đinh Hải Anh	Xem và sửa BTL	
3	Cài đặt thuật toán của bài toán tìm đường đi ngắn nhất	10/05/2025	Hà Thành	Xem và sửa BTL	
4	Cài đặt thuật toán của bài toán tìm đường đi ngắn nhất	10/05/2025	Nguyễn Minh Ngọc	Xem và sửa BTL	
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	Nguyễn Đức Trung	4	1	1	
2	Đinh Hải Anh	1	1	1	
3	Hà Thành	1	1	1	
4	Nguyễn Minh Ngọc	1	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 6					
1. Tên lớp: 20242IT6094007 Khóa: 18					
2. Nhóm: 6					
3. Thời gian: 11/05/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Nguyễn Đức Trung					
6. Thành viên tham dự - Participants: Nguyễn Đức Trung, Đinh Hải Anh , Hà Thành , Nguyễn Minh Ngọc					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Nguyễn Đức Trung	✓			Done
2	Đinh Hải Anh	✓			Done
3	Hà Thành	✓			Done
4	Nguyễn Minh Ngọc	✓			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	- Nhóm trưởng	
2	Thiết lập kênh giao tiếp, lưu trữ	Nguyễn Đức Trung	5p	- 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	- 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	Xác định nội dung cần làm	Cả nhóm	30p	Thống nhất nội dung cần làm
5	Phân chia công việc cho từng thành viên	Nguyễn Đức Trung	15p	Xem và sửa BTL
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 tìm kiếm mù, thuật toán tìm kiếm heuristic	thuật toán heuristic	Đề tài ứng dụng thuật toán tìm kiếm heuristic vào bài toán tìm đường đi ngắn nhất
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	Cài đặt thuật toán của bài toán tìm đường đi ngắn nhất	17/05/2025	Nguyễn Đức Trung	Xem và sửa BTL
2	Cài đặt thuật toán của bài toán tìm đường đi ngắn nhất	17/05/2025	Đình Hải Anh	Xem và sửa BTL
3	Cài đặt thuật toán của bài toán tìm đường đi ngắn nhất	17/05/2025	Hà Thành	Xem và sửa BTL
4	Cài đặt thuật toán của bài toán tìm đường đi ngắn nhất	17/05/2025	Nguyễn Minh Ngọc	Xem và sửa BTL

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	Nguyễn Đức Trung	4	1	1	
2	Đình Hải Anh	1	1	1	
3	Hà Thành	1	1	1	
4	Nguyễn Minh Ngọc	1	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 7					
1. Tên lớp: 20242IT6094007 Khóa: 18					
2. Nhóm: 6					
3. Thời gian: 18/05/2025					
4. Địa điểm: Google meet					
5. Người chủ trì cuộc họp: Nguyễn Đức Trung					
6. Thành viên tham dự - Participants: Nguyễn Đức Trung, Đinh Hải Anh , Hà Thành , Nguyễn Minh Ngọc					
#	Thành viên	Đúng giờ - On time	Muộn - Late	Vắng - Absent	Ghi chú
1	Nguyễn Đức Trung	✓			Done
2	Đinh Hải Anh	✓			Done
3	Hà Thành	✓			Done
4	Nguyễn Minh Ngọc	✓			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	- Nhóm trưởng	
2	Thiết lập kênh giao tiếp, lưu trữ	Nguyễn Đức Trung	5p	- 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	- 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	Xác định nội dung cần làm	Cả nhóm	30p	Thống nhất nội dung cần làm
5	Phân chia công việc cho từng thành viên	Nguyễn Đức Trung	15p	Xem và sửa BTL

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 tìm kiếm mù, thuật toán tìm kiếm heuristic	thuật toán heuristic	Đề tài ứng dụng thuật toán tìm kiếm heuristic vào bài toán tìm đường đi ngắn nhất
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	Hoàn thiện BTL	25/05/2025	Nguyễn Đức Trung	Hoàn thiện BTL	
2	Hoàn thiện BTL	25/05/2025	Đinh Hải Anh	Hoàn thiện BTL	
3	Hoàn thiện BTL	25/05/2025	Hà Thành	Hoàn thiện BTL	
4	Hoàn thiện BTL	25/05/2025	Nguyễn Minh Ngọc	Hoàn thiện BTL	
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	Nguyễn Đức Trung	4	1	1	
2	Đinh Hải Anh	1	1	1	
3	Hà Thành	1	1	1	
4	Nguyễn Minh Ngọc	1	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 nói đầu

Trong kỷ nguyên số, việc giải quyết các bài toán tối ưu hóa đóng vai trò then chốt trong sự phát triển của công nghệ và các hệ thống thông minh. Trong đó, bài toán tìm đường đi ngắn nhất không chỉ là một vấn đề lý thuyết mà còn có ý nghĩa thực tiễn sâu rộng, ứng dụng trong hàng loạt lĩnh vực như giao thông vận tải, mạng máy tính, robot tự hành, và các hệ thống định tuyến tối ưu

Các thuật toán tìm kiếm heuristics đã được chứng minh là công cụ đắc lực trong việc giải quyết các bài toán phức tạp này, nhờ khả năng dự đoán và định hướng tìm kiếm hợp lý, giúp giảm thiểu đáng kể thời gian xử lý và nâng cao hiệu quả tìm kiếm. Báo cáo này sẽ trình bày một cách khái quát về các thuật toán tìm kiếm heuristics, phân tích nguyên lý hoạt động, đồng thời khảo sát các ứng dụng của chúng trong bài toán tìm đường đi ngắn nhất – một trong những bài toán kinh điển của lĩnh vực này.

Thông qua nội dung nghiên cứu này, chúng tôi hy vọng độc giả sẽ có cái nhìn toàn diện hơn về các thuật toán heuristics, từ đó ứng dụng hiệu quả vào thực tiễn, góp phần thúc đẩy các giải pháp tối ưu hóa trong công nghệ thông tin và các hệ thống tự động ngày càng phát triển.

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 chân thành cảm ơn!

Mục Lục

BÁO CÁO HỌC TẬP NHÓM.....	1
PHIẾU HỌC TẬP NHÓM.....	4
BIÊN BẢN HỌP NHÓM.....	6
DANH MỤC HÌNH ẢNH	31
1.1.Không gian trạng thái	32
1.1.1 Mô tả trạng thái.	32
1.1.2. Toán tử chuyển trạng thái.	32
1.1.3. Không gian trạng thái của bài toán.....	33
1.2.Thuật toán tìm kiếm heuristic	34
1.2.1.Tổng quan về giải thuật tìm kiếm Heuristic.....	34
1.2.2.Thuật toán tìm kiếm heuristic	36
CHƯƠNG 2: ỨNG DỤNG THUẬT TOÁN HEURISTIC VÀO BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT	50
2.1. Không gian trạng thái của bài toán tìm đi ngắn nhất.	50
2.1.1.Giới thiệu bài toán.	50
2.2. Phân tích các thành phần để cài đặt thuật toán giải quyết bài toán.....	51
2.2.1. Giải thuật sử dụng.....	51
2.2.2. Đưa sơ đồ về dạng danh sách kề.....	52
2.2.3. Xây dựng bài toán.....	52
2.2.4. Cài đặt chi tiết	55
KẾT LUẬN	60

DANH MỤC HÌNH ẢNH

Hình 1.1. Đồ thị minh họa trong tìm kiếm tối ưu

Hình 1.2 : Đồ thị minh họa thuật toán A

Hình 1.3 : Đồ thị minh họa thuật toán A^{KT}

Hình 1.4 : Thuật giải bài trò chơi 8 ô chữ của A^{KT}

Hình 1.5 : Đồ thị minh họa thuật toán A^*

Hình 2.1 : Tìm đường đi ngắn nhất từ Bắc Ninh đến Nam Định

Hình 2.2: Dạng danh sách kề của đồ thị

Hình 2.3: Hàm trả về giá trị heuristic cho nút

Hình 2.4: hàm in ra đường đi

Hình 2.5 : Hàm xây dựng đường đi

Hình 2.6 : Chương trình chạy ra đường đi ngắn nhất

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

1.1. Không gian trạng thái

1.1.1 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.

tình trạng cuối của bài toán gọi là trạng thái đầu và trạng thái cuối.

Ví dụ: Bài toán đong nước: Cho 2 bình có dung tích lần lượt là m và n (lit). Với nguồn nước không hạn chế, dùng 2 bình trên để đong k lít nước. Không mất

tính tổng quát có thể giả thiết $k \leq \min(m, n)$.

- Tại mỗi thời điểm xác định, lượng nước hiện có trong mỗi bình phản ánh

bản chất hình trạng của bài toán ở thời điểm đó. - Gọi x là lượng nước hiện có trong bình dung tích m và y là lượng nước hiện có

trong bình dung tích n. - Như vậy bộ có thứ tự (x,y) có thể xem là trạng thái của bài toán. Với cách

mô tả như vậy, các trạng thái đặc biệt của bài toán sẽ là:

+ Trạng thái đầu: (0,0)

+ Trạng thái cuối: (x,k) hoặc (k,y)

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 để biểu diễn các toán tử:

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

Ví dụ 1. Bài toán đong nước

Các thao tác sử dụng để chuyển trạng thái này sang trạng thái khác gồm:

+ Đổ đầy một bình

+ Đổ hết nước trong một bình ra ngoài

+ Đổ nước từ bình này sang bình khác. Như vậy, nếu trạng thái đang xét là (x,y) thì các trạng thái kế tiếp có thể

chuyển đến sẽ là: - Đổ đầy bình thứ nhất (m) :

→ (m, y)

- Đổ đầy bình thứ hai (n) :

→ (x, n)

- Đổ hết nước trong bình thứ nhất:

→ $(0, y)$

- Đổ hết nước trong bình thứ hai:

→ $(x, 0)$

- Đổ nước từ bình thứ nhất sang bình thứ hai:

+ Nếu $x + y \leq n \rightarrow (0, x + y)$

+ Nếu $x + y > n \rightarrow (x - (n - y), n)$

- Đổ nước từ bình thứ hai sang bình thứ nhất:

+ Nếu $x + y \leq m \rightarrow (x + y, 0)$

+ Nếu $x + y > m \rightarrow (m, y - (m - x))$

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 : Không gian trạng thái của bài toán đong nước là bộ bốn T, S, G, F xác định như sau:

$$T = \{(x,y) / 0 \leq x \leq m; 0 \leq y \leq n\}$$

$$S = (0,0)$$

$$G = \{(x,k) \text{ hoặc } (k,y) / 0 \leq x \leq m; 0 \leq y \leq n\}$$

F = Tập các thao tác đong đầy, đổ ra hoặc đổ sang bình khác thực hiện trên một bình.

1.2. Thuật toán tìm kiếm heuristic

1.2.1. Tổng quan về giải thuật tìm kiếm Heuristic.

1.2.1.1. Khái niệm

Tìm kiếm heuristic là một phương pháp tìm kiếm có sử dụng thông tin bổ sung để đánh giá và lựa chọn các trạng thái tiềm năng trong không gian tìm kiếm. Không giống như tìm kiếm mù, vốn mở rộng trạng thái theo một quy tắc cố định (như BFS hoặc DFS), tìm kiếm heuristic dựa vào một hàm heuristic để ước lượng chi phí hoặc lợi ích của một bước đi nhất định, giúp tăng tốc độ tìm kiếm lời giải.

- Heuristic chỉ là một phỏng đoán chứa các thông tin về bước tiếp theo sẽ được chọn dùng trong việc giải quyết một vấn đề.

- Heuristic là những tri thức được rút ra từ những kinh nghiệm, “trực giác” của con người

- Heuristic có thể là những tri thức đúng hoặc sai

Vì các heuristic sử dụng những thông tin hạn chế nên chúng ít khi có khả năng đoán trước chính xác cách hành xử của không gian trạng thái ở những giai đoạn xa hơn

1.2.1.2. Chức năng của Heuristic.

Các chương trình giải quyết những vấn đề trí tuệ nhân tạo sử dụng Heuristic cơ bản theo hai dạng:

- Vấn đề có thể không có giải pháp chính xác vì những điều không rõ ràng trong diễn đạt vấn đề hoặc trong các dữ liệu có sẵn.

- Vấn đề có thể có giải pháp chính xác, nhưng chi phí tính toán để tìm ra nó không cho phép.

1.2.1.3. Ưu điểm của Heuristic.

Thuật giải Heuristic thể hiện cách giải bài toán với các đặc tính sau:

- Thường tìm được lời giải tốt (Nhưng không chắc là lời giải tốt nhất).

- Giải bài toán theo thuật giải Heuristic thường dễ dàng và nhanh chóng đưa ra kết quả hơn so với giải thuật tối ưu, vì vậy chi phí thấp hơn.

- Thuật giải Heuristic thường thể hiện khá tự nhiên, gần gũi với cách suy nghĩ và hành động con người.

1.2.1.4. Thuật giải Heuristic:

- Heuristic là những tri thức được rút ra từ kinh nghiệm con người, nó có thể đúng hoặc sai.

- Heuristic thường được dùng trong các trường hợp sau:

- + Bài toán có thể không có nghiệm chính xác do các mệnh đề không phát biểu chặt chẽ hay thiếu dữ liệu để khẳng định kết quả.

- + Bài toán có nghiệm chính xác nhưng phí tổn tính toán để đưa ra nghiệm là quá lớn (bùng nổ tổ hợp).

- Thuật giải Heuristic là mở rộng khái niệm thuật toán và có đặc điểm:

- + Thường tìm lời giải tốt nhưng không tốt nhất.

- + Nhanh chóng tìm ra được giải thuật tối ưu, tiết kiệm chi phí hơn.

- + Thường thể hiện khá tự nhiên, gần gũi với cách hành động và suy nghĩ của con người.

1.2.1.5. Thiết kế thuật giải heuristic:

Có nhiều phương pháp để thiết kế thuật giải Heuristic, trong đó người ta sử dụng 2 nguyên lý cơ bản sau:

-Nguyên lý vét cạn thông minh: Khi không gian tìm kiếm D lớn, ta thường tìm cách giới hạn lại không gian tìm kiếm này hoặc thực hiện một kiểu dò tìm đặc biệt dựa vào đặc thù của bài toán để nhanh chóng tìm ra mục tiêu.

-Nguyên lý tham lam: Lấy tiêu chuẩn tối ưu nhất trên phạm vi toàn cục của bài toán để làm tiêu chuẩn chọn lựa hành động cho phạm vi cục bộ của từng bước trong quá trình tìm kiếm lời giải.

-Nguyên lý thứ tự: Thực hiện hành động dựa trên một cấu trúc thứ tự hợp lý của không gian khảo sát nhằm nhanh chóng đạt được lời giải tốt.

-Tìm kiếm leo đồi: Tìm kiếm leo đồi (HCS) thực chất là tìm kiếm theo chiều sâu nhưng không thể quay lui. Trong tìm kiếm leo đồi, việc lựa chọn trạng thái tiếp theo được quyết định dựa trên một hàm Heuristic:

Tư tưởng của thuật giải được thể hiện qua 2 bước:

i, Nếu trạng thái bắt đầu cũng là trạng thái đích thì thoát và báo là tìm được lời giải.

ii, Lặp lại cho đến khi trạng thái kết thúc goặc cho đến khi T_i không tồn tại 1 trạng thái nào tốt hơn trạng thái hiện tại.

+ Đặt S bằng tập tất cả các trạng thái có thể có của T_i và tốt hơn T_i .

+ Xác định T_{max} là trạng thái tốt nhất trong tập S : $T_i = T_{max}$

Và Lặp lại cho đến khi báo dừng.

1.2.2. Thuật toán tìm kiếm heuristic

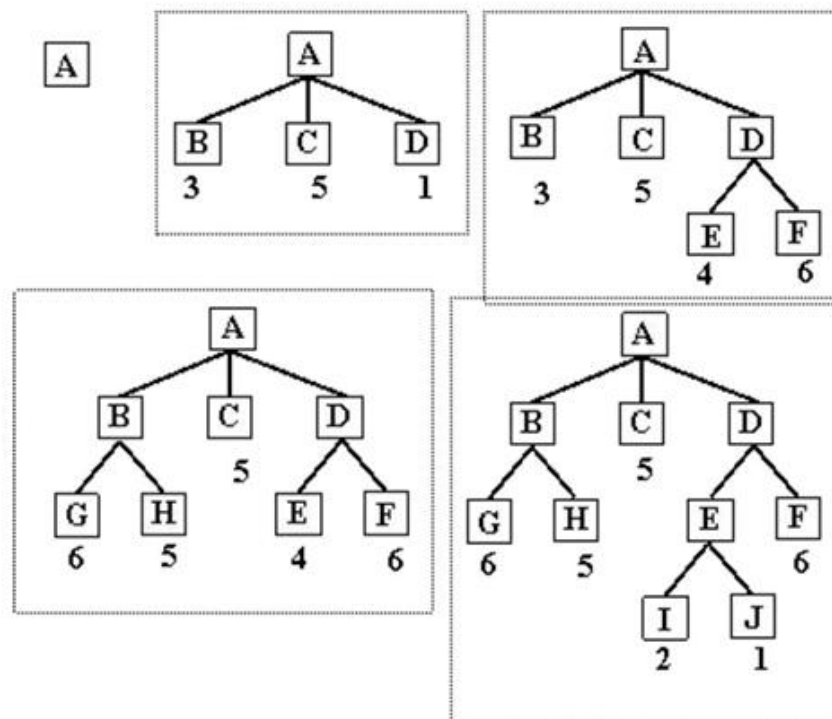
1.2.2.1. Tìm kiếm tối ưu (Best-First-Search)

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-BeFS) 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 BeFS, ta chọn đi theo trạng thái có khả năng cao nhất trong số các trạng thái đã được xét cho đến thời điểm đó. BeFS 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 từ 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 1.1. Đồ thị minh họa trong 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 (Tmax) trong OPEN (và xóa Tmax khỏi OPEN)

- + Nếu Tmax là trạng thái kết thúc thì thoát.

- + Ngược lại, tạo ra các trạng thái kế tiếp Tk có thể có từ trạng thái Tmax.

Đối với mỗi trạng thái kế tiếp

Tk thực hiện: Tính $f(T_k)$;

Thêm Tk 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 BeFS, 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:

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

1.2.2.2. Thuật toán A^T

Thuật giải A^T là một phương pháp tìm kiếm theo kiểu BeFS 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).

Giải thuật :

1. Đặt OPEN chứa trạng thái khởi đầu.
2. Cho đến khi tìm được trạng thái đích hoặc không còn nút nào trong OPEN, thực hiện:
 - a. Chọn trạng thái (T_{max}) có giá trị g nhỏ nhất trong OPEN (và xóa T_{max} khỏi OPEN)
 - b. Nếu T_{max} là trạng thái kết thúc thì thoát.
 - c. 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:

$$g(T_k) = g(T_{max}) + \text{cost}(T_{max}, T_k)$$

Thêm T_k vào OPEN.

***Note:** Vì chỉ sử dụng hàm g (mà không dùng hàm ước lượng h' để đánh giá độ tốt của một trạng thái nên ta cũng có thể xem A^T chỉ là một thuật toán.

Thuật toán

Vào: Đồ thị $G = (V, E)$

$C: E \rightarrow \mathbb{R}^+$

$e \mapsto C(e)$

Đỉnh đầu T_0 và Goal chứa tập các đỉnh đích

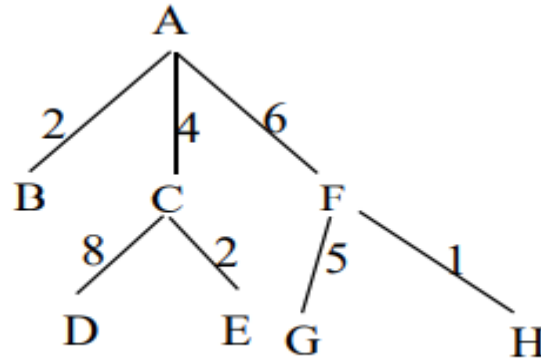
Ra: Đường đi $p: T_0 \rightarrow TG \in \text{Goal}$ sao cho:

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

Phương pháp: Sử dụng hai danh sách DONG và MO

```
void AT(){
    MO = {T0}; g(T0) = 0; DONG = ∅;
    while (MO != ∅){
        n = getNew(MO) // Lấy đỉnh n sao cho g(n) -> min
        if (n == TG) return True;
        else{
            for (m ∈ B(n))
                if (m ∉ MO) && (m ∉ DONG){
                    g(m)=g(n) + cost(m,n); MO=MO ∪ {m};
                }
            else g(m)=min{g(m), gnew(m)};
            DONG = DONG ∪ {n};
        }
    }
    return False;
}
```

Ví dụ:



Hình 1.2 : Đồ thị minh họa thuật toán A^T

Có thể trình bày quá trình tìm kiếm bằng bảng dưới đây. Ký hiệu giá trị $g(n)$ là chỉ số dưới tương ứng đỉnh n : $g(n)$

I	T(i)	OPEN	CLOSE
		A(0)	
A	B,C,D	B (8), C (4), D (5)	A
C	G	B (8), D (5), G (5)	A, C
D	H,I	B (8), G (5), H (14), I (6)	A, C, D
G		B (8), H (14), I (6)	A, C, D, G
I	K	B (8), H (14), K (8)	A, C, D, G, I
B	E,F	H (14), K (8), E (10), F (11)	A, C, D, G, I, B
K			

Trong bảng trên các con số trong ngoặc đơn là giá trị của hàm $g()$. Do K

\in Goal nên thuật toán dừng và đường đi tìm được là:

$A \rightarrow D \rightarrow I \rightarrow K$ với $C(p) = 8$

Kết quả:

Nếu trong đồ thị G tồn tại đường đi $p: T_0 \rightarrow T_G \in \text{Goal}$ thì thuật toán 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 BeFS

+ 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

1.2.2.3. Thuật toán A^{KT}

Thuật giải A^{KT} 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 $f(n)$ 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 sung 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.

Giải thuật :

1. Đặt OPEN chứa trạng thái khởi đầu.

2. Cho đến khi tìm được trạng thái đích hoặc không còn nút nào trong OPEN, thực hiện :

- a. Chọn trạng thái (Tmax) có giá trị f nhỏ nhất trong OPEN (và xóa Tmax khỏi OPEN)
- b. Nếu Tmax là trạng thái kết thúc thì thoát.
- c. Ngược lại, tạo ra các trạng thái kế tiếp Tk có thể có từ trạng thái Tmax. Đối với mỗi trạng thái kế tiếp Tk thực hiện :

$$g(Tk) = g(Tmax) + \text{cost}(Tmax, Tk).$$

Tính $h'(Tk)$.

$$f(Tk) = g(Tk) + h'(Tk) .$$

Thêm Tk vào OPEN.

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 T0 và tập các đỉnh đích

Ra: Đường đi p: $T0 \rightarrow TG \in \text{Goal}$

Phương pháp: Sử dụng 2 danh sách DONG và MO

void AKT () {

MO = {T0}, $g(T0) = 0$;

Tính $h(T0)$, $f(T0) = g(T0) + h(T0)$

while (MO $\neq \emptyset$) {

n = getNew(MO) // lấy đỉnh n sao cho $f(n)$ đạt min

if (n == TG) return True;

else {

for (m B(n)) {

```

    g(m) = g(n) + cost(m,n);
    Tính h(m), f(m) = g(m) + h(m);
    OM = MO ∪ {m};
}
}
}
return False;
}

```

Ví dụ:

Trạng thái đầu

2	8	3
1	6	4
7	■	5

Trạng thái đích

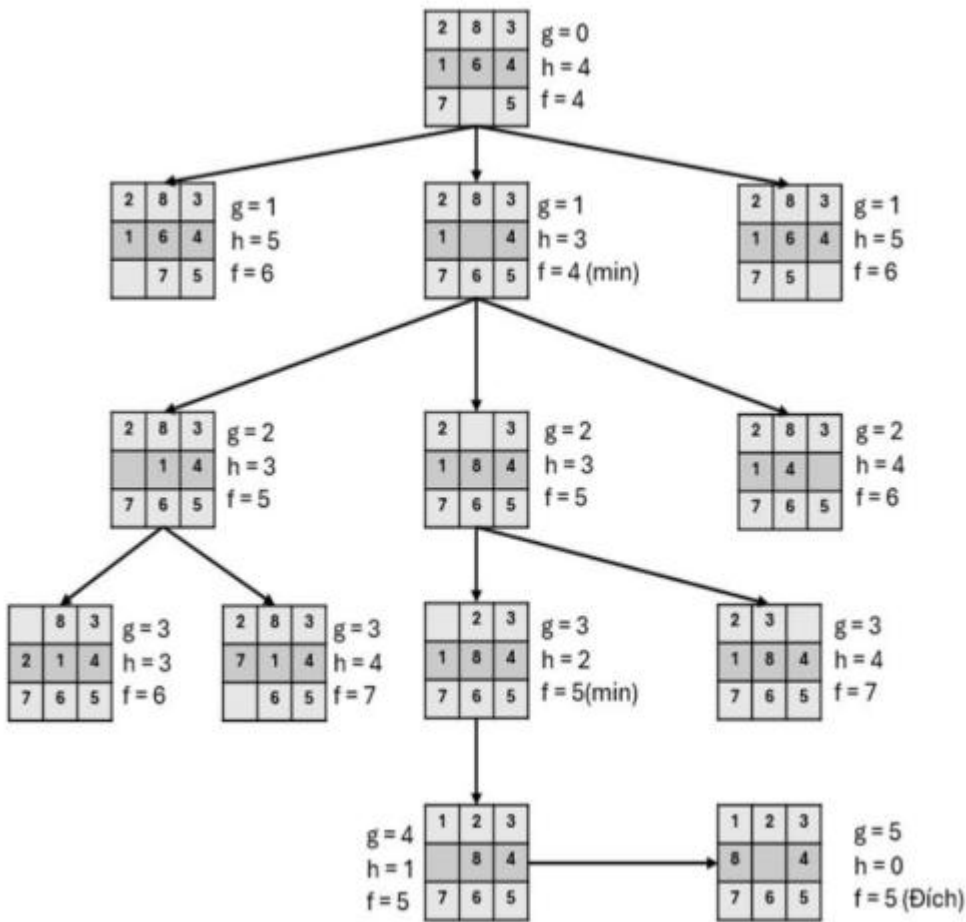
1	2	3
8	■	4
7	6	5

Hình 1.3 : Đồ thị minh họa thuật toán A^{KT}

Chọn hàm $f(n) = g(n) + h(n)$

Trong đó:

+ $g(n)$ là giá của đường đi hiện tại từ đỉnh T_0 tới đỉnh n (số lần dịch chuyển ô trống từ trạng thái s đến trạng thái n). + $h(n)$: số các con số không nằm đúng vị trí của nó so với trạng thái đích. Chẳng hạn, với bài toán này, khi xét trạng thái ban đầu thì $g(n)=0$ (do ô trống chưa dịch chuyển lần nào) và $h(n) = 4$ suy ra $f(n) = 4$. Trong hình vẽ là kết quả áp dụng thủ tục A^{KT} với tri thức bổ sung $h(n)$, giá trị của f đối với mỗi nút được cho trong hình tròn.



Hình 1.4 : Thuật giải bài trò chơi 8 ô chữ của A^{KT}

1.2.2.4. Thuật toán A^*

1.2.2.4.1. Khái niệm

A^* là giải thuật tìm kiếm trong đồ thị, tìm đường đi từ một đỉnh hiện tại đến đỉnh đích có sử dụng hàm để ước lượng khoảng cách hay còn gọi là hàm Heuristic.

Từ trạng thái hiện tại A^* xây dựng tất cả các đường đi có thể đi dùng hàm ước lượng khoảng cách (hàm Heuristic) để đánh giá đường đi tốt nhất có thể đi. Tùy theo mỗi dạng bài khác nhau mà hàm Heuristic sẽ được đánh giá khác nhau. A^* luôn tìm được đường đi ngắn nhất nếu tồn tại đường đi như thế.

A* lưu giữ một tập các đường đi qua đồ thị, từ đỉnh bắt đầu đến đỉnh kết thúc, tập các đỉnh có thể đi tiếp được lưu trong tập OPEN.

Thứ tự ưu tiên cho một đường đi được quyết định bởi hàm Heuristic được đánh giá

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

Trong đó:

- $g(x)$ là chi phí của đường đi từ điểm xuất phát cho đến thời điểm hiện tại.
- $h(x)$ là hàm ước lượng chi phí từ đỉnh hiện tại đến đỉnh đích $f(x)$ thường có giá trị càng thấp thì độ ưu tiên càng cao.
- Open: tập các trạng thái đã được sinh ra nhưng chưa được xét đến.

1.2.2.4.2. Giải thuật :

- Close: tập các trạng thái đã được xét đến.
- $\text{Cost}(p, q)$: là khoảng cách giữa p, q .
- $g(p)$: khoảng cách từ trạng thái đầu đến trạng thái hiện tại p .
- $h(p)$: giá trị được lượng giá từ trạng thái hiện tại đến trạng thái đích.
- $f(p) = g(p) + h(p)$

Các bước thực hiện :

- Bước 1:

Open: = {s}

Close: = {}

- Bước 2: while (Open != {})

+ Chọn trạng thái (đỉnh) tốt nhất p trong Open (xóa p khỏi Open).

+ Nếu p là trạng thái kết thúc thì thoát.

+ Chuyển p qua Close và tạo ra các trạng thái kế tiếp q sau p .

Nếu q đã có trong Open

- Nếu $g(q) > g(p) + \text{Cost}(p, q)$ thì

$$g(q) = g(p) + \text{Cost}(p, q)$$

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

$\text{prev}(q) = p$ (đỉnh cha của q là p)

Nếu q chưa có trong Open

$$g(q) = g(p) + \text{cost}(p, q)$$

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

$\text{prev}(q) = p$

Thêm q vào Open

Nếu q có trong Close

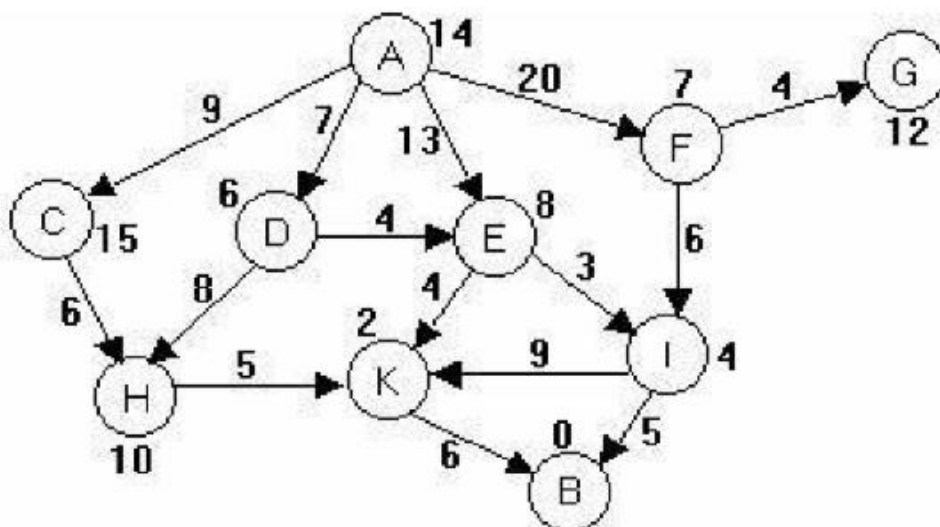
Nếu $g(q) > g(p) + \text{Cost}(p, q)$

Bỏ q khỏi Close

Thêm q vào Open

- Bước 3: Không tìm được.

Ví dụ :



Hình 1.5 : Đồ thị minh họa thuật toán 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:

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

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

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):

$$g(H) = g(D) + \text{Độ dài cung (D, H)} = 7 + 8 = 15, f(H) = 15 + 10 = 25. \text{ Đường đi tới E qua D có độ dài:}$$

$$g(E) = g(D) + \text{Độ dài cung (D, E)} = 7 + 4 = 11. \text{ Vậy đỉnh E mới có đánh giá là } f(E) = g(E) + h(E) = 11 + 8 = 19. \text{ Trong số các đỉnh cho phát triển, thì đỉnh E với đánh giá } f(E) = 19 \text{ là đỉnh}$$

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

1.2.2.4.3. Các tính chất :

Cũng như tìm kiếm theo chiều rộng (breadth-first search), A^* là thuật toán đầy đủ (complete) theo nghĩa rằng nó sẽ luôn luôn tìm thấy một lời giải nếu bài toán có lời giải.

Nếu hàm heuristic h có tính chất thu nạp được (admissible), nghĩa là nó không bao giờ đánh giá cao hơn chi phí nhỏ nhất thực sự của việc đi tới đích, thì bản thân A^* có tính chất thu nạp được (hay tối ưu) nếu sử dụng một tập đóng. Nếu không sử dụng tập đóng thì hàm h phải có tính chất đơn điệu (hay nhất quán) thì A^* mới có tính chất tối ưu. Nghĩa là nó không bao giờ đánh giá chi phí đi từ một nút tới một nút kế nó cao hơn chi phí thực. Phát biểu một cách hình thức, với mọi nút x, y trong đó y là nút tiếp theo của x :

A* còn có tính chất hiệu quả một cách tối ưu (optimally efficient) với mọi hàm heuristic h , có nghĩa là không có thuật toán nào cũng sử dụng hàm heuristic đó mà chỉ phải mở rộng ít nút hơn A*, trừ khi có một số lời giải chưa đầy đủ mà tại đó h dự đoán chính xác chi phí của đường đi tối ưu.

Quan hệ với tìm kiếm chi phí đều (uniform-cost search)

Thuật toán Dijkstra là một trường hợp đặc biệt của A* trong đó đánh giá heuristic là một hàm hằng $h(x) = 0$ với mọi x .

Mức độ phức tạp

Vấn đề sử dụng bộ nhớ của A* còn rắc rối hơn độ phức tạp thời gian. Trong trường hợp xấu nhất, A* phải ghi nhớ số lượng nút tăng theo hàm mũ. Một số biến thể của A* đã được phát triển để đối phó với hiện tượng này, một trong số đó là A* lặp sâu dần (iterative deepening A*), A* bộ nhớ giới hạn (memory-bounded A*- MA*) và A* bộ nhớ giới hạn đơn giản (simplified memory bounded A*). Một thuật toán tìm kiếm có thông tin khác cũng có tính chất tối ưu và đầy đủ nếu đánh giá heuristic của nó là thu nạp được (admissible). Đó là tìm kiếm đệ quy theo lựa chọn tốt nhất (recursive best-first search- RBFS).

Ưu điểm

Một thuật giải linh động, tổng quát, trong đó hàm chứa cả tìm kiếm chiều sâu, tìm kiếm chiều rộng và những nguyên lý Heuristic khác. Nhanh chóng tìm đến lời giải với sự định hướng của hàm Heuristic. Chính vì thế mà người ta thường nói A* chính là thuật giải tiêu biểu cho Heuristic.

Nhược điểm

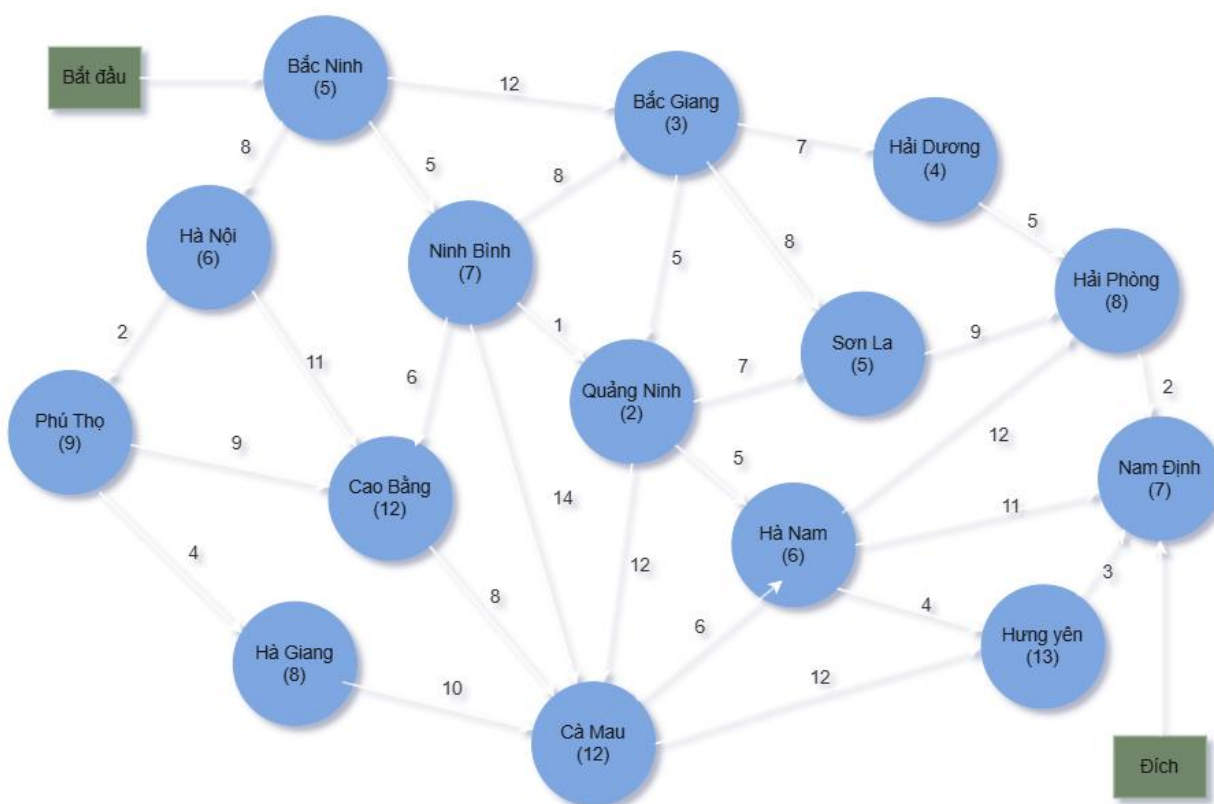
A* rất linh động nhưng vẫn gặp một khuyết điểm cơ bản- giống như chiến lược tìm kiếm chiều rộng- đó là tốn khá nhiều bộ nhớ để lưu lại những trạng thái đã đi qua

CHƯƠNG 2: ỨNG DỤNG THUẬT TOÁN HEURISTIC VÀO BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT

2.1. Không gian trạng thái của bài toán tìm đi ngắn nhất.

2.1.1. Giới thiệu bài toán.

Một du khách có trong tay bản đồ giao thông nối các tỉnh của nước Việt Nam. Du khách đang ở Bắc Ninh và anh ta muốn tìm đường đi nhanh nhất tới Nam Định.



Hình 2.1 : Tìm đường đi ngắn nhất từ Bắc Ninh đến Nam Định

Trong bài toán này, các tỉnh thành có trong bản đồ là các trạng thái tỉnh Bắc Ninh là trạng thái đầu còn Nam Định là trạng thái kết thúc. Với bài toán này ta có thể sử dụng cách biểu diễn của đồ thị (ma trận kề, ma trận trọng số, danh sách cạnh, danh sách kề).

2.2. Phân tích các thành phần để cài đặt thuật toán giải quyết bài toán

2.2.1. Giải thuật sử dụng

A* là giải thuật tìm kiếm trong đồ thị, tìm đường đi từ một đỉnh hiện tại đến đỉnh đích có sử dụng hàm để ước lượng khoảng cách hay còn gọi là hàm Heuristic.

Từ trạng thái hiện tại A* xây dựng tất cả các đường đi có thể đi dùng hàm ước lượng khoảng cách (hàm Heuristic) để đánh giá đường đi tốt nhất có thể đi. Tùy theo mỗi dạng bài khác nhau mà hàm Heuristic sẽ được đánh giá khác nhau. A* luôn tìm được đường đi ngắn nhất nếu tồn tại đường đi như thế.

A* lưu giữ một tập các đường đi qua đồ thị, từ đỉnh bắt đầu đến đỉnh kết thúc, tập các đỉnh có thể đi tiếp được lưu trong tập Open.

Thứ tự ưu tiên cho một đường đi được quyết định bởi hàm Heuristic được đánh giá $f(x) = g(x) + h(x)$

- $g(x)$ là chi phí của đường đi từ điểm xuất phát cho đến thời điểm hiện tại.
- $h(x)$ là hàm ước lượng chi phí từ đỉnh hiện tại đến đỉnh đích $f(x)$ thường có giá trị càng thấp thì độ ưu tiên càng cao.

2.2.2. Đưa sơ đồ về dạng danh sách kề

```
graph = {
    'Bắc Ninh': [('Hà Nội', 8), ('Ninh Bình', 5), ('Bắc Giang', 12)],
    'Hà Nội': [('Phú Thọ', 2), ('Cao Bằng', 11)],
    'Phú Thọ': [('Hà Giang', 4), ('Cao Bằng', 9)],
    'Hà Giang': [('Cà Mau', 10)],
    'Cao Bằng': [('Cà Mau', 8)],
    'Ninh Bình': [('Cao Bằng', 6), ('Bắc Giang', 8), ('Quảng Ninh', 1)],
    'Bắc Giang': [('Hải Dương', 7), ('Sơn La', 8)],
    'Hải Dương': [('Hải Phòng', 5)],
    'Hải Phòng': [('Nam Định', 2)],
    'Quảng Ninh': [('Bắc Giang', 5), ('Sơn La', 7), ('Hà Nam', 5)],
    'Sơn La': [('Hải Phòng', 9)],
    'Cà Mau': [('Hà Nam', 6), ('Hưng yên', 12)],
    'Hà Nam': [('Nam Định', 12), ('Hưng yên', 4)],
    'Hưng yên': [('Nam Định', 3)],
    'Nam Định': []
}
```

Hình 2.2: Dạng danh sách kề của đồ thị

2.2.3. Xây dựng bài toán

Ghi chú quy ước:

- $g(n)$: tổng chi phí từ Bắc Ninh đến đỉnh n
- $h(n)$: giá trị heuristic (số trong ngoặc)
- $f(n) = g(n) + h(n)$
- OPEN: hàng đợi ưu tiên (ưu tiên đỉnh có f nhỏ nhất)
- CLOSE: tập các đỉnh đã xét

OPEN = [Bắc Ninh, $g=0$, $h=5$, $f=5$]

CLOSE = []

→ Lấy Bắc Ninh ra xử lý.

Đỉnh kề của Bắc Ninh:

- Bắc Giang: $g = 0 + 12 = 12$, $h = 3 \rightarrow f = 15$
- Ninh Bình: $g = 0 + 5 = 5$, $h = 7 \rightarrow f = 12$
- Hà Nội: $g = 0 + 8 = 8$, $h = 6 \rightarrow f = 14$

OPEN = [Ninh Bình($f=12$), Hà Nội($f=14$), Bắc Giang($f=15$)]

CLOSE = [Bắc Ninh]

Lấy Ninh Bình ($f=12$) ra.

Đỉnh kề của Ninh Bình:

- Bắc Giang (đã trong OPEN): $g = 5 + 8 = 13$, $f = 13 + 3 = 16 \rightarrow$ không tối ưu hơn Bắc Giang($f=15$), bỏ qua
- Quảng Ninh: $g = 5 + 1 = 6$, $h = 2 \rightarrow f = 8$
- Cà Mau: $g = 5 + 14 = 19$, $h = 12 \rightarrow f = 31$
- Cao Bằng: $g = 5 + 6 = 11$, $h = 12 \rightarrow f = 23$

OPEN = [Quảng Ninh($f=8$), Hà Nội($f=14$), Bắc Giang($f=15$), Cà Mau($f=31$), Cao Bằng($f=23$)]

CLOSE = [Bắc Ninh, Ninh Bình]

Lấy Quảng Ninh ($f=8$) ra.

Đỉnh kề:

- Bắc Giang: $g = 6 + 5 = 11$, $f = 14 \rightarrow$ cập nhật Bắc Giang vì $g=11 < g=12$ trước đó
- Sơn La: $g = 6 + 7 = 13$, $h = 5 \rightarrow f = 18$
- Hà Nam: $g = 6 + 5 = 11$, $h = 6 \rightarrow f = 17$

OPEN = [Bắc Giang(f=14), Hà Nội(f=14), Cao Bằng(f=23), Cà Mau(f=31),
Sơn La(f=18), Hà Nam(f=17)]

CLOSE = [Bắc Ninh, Ninh Bình, Quảng Ninh]

Lấy Hà Nội (f=14) ra.

Đỉnh kề:

- Cao Bằng: $g = 8 + 11 = 19$, $f = 19 + 12 = 31$ (không tối ưu hơn đã có)
- Phú Thọ: $g = 8 + 2 = 10$, $h = 9 \rightarrow f = 19$

OPEN = [Bắc Giang(f=14), Hà Nam(f=17), Sơn La(f=18), Phú Thọ(f=19),
Cao Bằng(f=23), Cà Mau(f=31)]

CLOSE = [Bắc Ninh, Ninh Bình, Quảng Ninh, Hà Nội]

Lấy Bắc Giang (f=14) ra.

Đỉnh kề:

- Hải Dương: $g = 11 + 7 = 18$, $h = 4 \rightarrow f = 22$
- Sơn La: $g = 11 + 8 = 19$, $f = 24$

OPEN = [Hà Nam(f=17), Sơn La(f=18), Phú Thọ(f=19), Hải Dương(f=22),
Cao Bằng(f=23), Cà Mau(f=31)]

CLOSE = [Bắc Ninh, Ninh Bình, Quảng Ninh, Hà Nội, Bắc Giang]

Lấy Hà Nam (f=17) ra.

Đỉnh kề:

- Nam Định (đích!): $g = 11 + 12 = 23$, $h = 7 \rightarrow f = 30$
- Hưng Yên: $g = 11 + 4 = 15$, $h = 13 \rightarrow f = 28$

OPEN = [Sơn La(f=18), Phú Thọ(f=19), Hải Dương(f=22), Cao Bằng(f=23),
Hưng Yên(f=28), Cà Mau(f=31), Nam Định(f=30)]

CLOSE = [Bắc Ninh, Ninh Bình, Quảng Ninh, Hà Nội, Bắc Giang, Hà Nam]

Lấy Sơn La (f=18) ra \rightarrow không dẫn đến đích nhanh hơn. Loại dần: Phú Thọ,
Hải Dương, Cao Bằng, không mang lại đường ngắn hơn. Lấy Hưng Yên
(f=28) ra.

Đỉnh kề:

- Nam Định: $g = 15 + 3 = 18$, $h = 7 \rightarrow f = 25$
 \rightarrow Cập nhật Nam Định: $g=18$, $f=25$

Lấy Nam Định ($f=25$) \rightarrow là đích. Đường đi ngắn nhất:

Bắc Ninh \rightarrow Ninh Bình \rightarrow Quảng Ninh \rightarrow Hà Nam \rightarrow Hưng Yên \rightarrow Nam Định

Tổng chi phí:

$$5 (BN \rightarrow NB) + 1 (NB \rightarrow QN) + 5 (QN \rightarrow HN) + 4 (HN \rightarrow HY) + 3 (HY \rightarrow ND) = 18$$

2.2.4. Cài đặt chi tiết

2.2.4.1. Hàm heuristic

```
18  def h(n): 3 usages
19      heuristic = {
20          'Bắc Ninh': 5, 'Bắc Giang': 3, 'Hải Dương': 4, 'Hải Phòng': 8,
21          'Hà Nội': 6, 'Phú Thọ': 9, 'Cao Bằng': 12, 'Hà Giang': 8,
22          'Ninh Bình': 7, 'Quảng Ninh': 2, 'Sơn La': 5, 'Hà Nam': 6,
23          'Nam Định': 7, 'Hưng yên': 13, 'Cà Mau': 12
24      }
25      return heuristic.get(n, float('inf'))
```

Hình 2.3: Hàm trả về giá trị heuristic cho nút

+ Hàm h này nhận vào một nút node và trả về giá trị heuristic (ước lượng chi phí từ nút đó đến đích) cho nút đó.

+ Dữ liệu heuristic được lưu trong từ điển heuristic_values. Nếu node có trong từ điển, hàm trả về giá trị heuristic tương ứng. Nếu không, nó trả về float('inf'), một giá trị vô hạn, biểu thị rằng không có thông tin heuristic cho nút đó.

2.2.4.2. Hàm in ra đường đi ngắn nhất

```
27  def print_path_and_cost(start, goal, parent, g): 1 usage
28      path = []
29      node = goal
30      while node in parent:
31          path.append(node)
32          node = parent[node]
33      path.append(start)
34      path.reverse()
35      print("\nĐường đi ngắn nhất:")
36      print(" -> ".join(path))
37      print("Tổng chi phí:", g[goal])
```

Hình 2.4: hàm in ra đường đi

- + Hàm này in ra **đường đi** từ nút start đến nút goal và **chi phí** tương ứng (có thể hiểu là chi phí từ start đến goal, được lưu trong `g[goal]`).
- + Để in ra đường đi, hàm lần lượt truy ngược các nút từ goal về start thông qua từ điển `parent`, giúp xác định "cha" của mỗi nút.
- + Sau khi có đường đi, hàm đảo ngược danh sách `path` để hiển thị đúng thứ tự từ start đến goal.
- + Cuối cùng, hàm in ra đường đi và chi phí $C(p)$ (là chi phí từ start đến goal).

2.2.4.3. Xây dựng hàm tìm đường đi ngắn nhất

```
39 def A_star(graph, start, goal): 1 usage
40     OPEN = [start]
41     CLOSE = []
42     g = {start: 0}
43     f = {start: h(start)}
44     parent = {}
45
46     while OPEN:
47         # Chọn node có f nhỏ nhất
48         n = min(OPEN, key=lambda x: f[x])
49         if n == goal:
50             print_path_and_cost(start, goal, parent, g)
51             return
52
53         OPEN.remove(n)
54         CLOSE.append(n)
55
56         for m, cost in graph.get(n, {}).items():
57             new_g = g[n] + cost
58             if m not in OPEN and m not in CLOSE:
59                 g[m] = new_g
60                 f[m] = g[m] + h(m)
61                 parent[m] = n
62                 OPEN.append(m)
63             elif m in OPEN and new_g < g[m]:
64                 g[m] = new_g
65                 f[m] = g[m] + h(m)
66                 parent[m] = n
67
68         print(f"\nXét định: {n}")
69         print("OPEN:")
70         for x in OPEN:
71             print(f"    {x}: g={g[x]}, f={f[x]}, cha={parent.get(x)}")
72         print("CLOSE:")
73         for x in CLOSE:
74             print(f"    {x}: g={g[x]}, f={f[x]}")
75
76     print("Không tìm thấy đường đi.")
```

Hình 2.5 : Hàm xây dựng đường đi

Hàm A_star là thuật toán A* để tìm đường đi ngắn nhất từ một đỉnh start đến một hoặc nhiều đỉnh đích trong đồ thị. Đoạn mã thực hiện các bước sau:

Khởi tạo:

- MO là tập đỉnh mở, chứa các đỉnh mà thuật toán chưa xét đến. Ban đầu chỉ chứa start.
- DONG là tập đỉnh đã xét.
- g là từ điển lưu chi phí từ start đến mỗi đỉnh. Ban đầu, $g[start] = 0$ (chi phí từ start đến chính nó là 0).
- f là từ điển lưu giá trị $f(n) = g(n) + h(n)$ (chi phí ước tính từ start qua nút n đến mục tiêu). Ban đầu, $f[start] = h(start)$ (giá trị heuristic cho start).
- parent là từ điển lưu trữ cha của mỗi đỉnh để theo dõi đường đi.

Quy trình thuật toán:

+ Chọn đỉnh n có giá trị $f(n)$ nhỏ nhất trong tập MO (đỉnh ưu tiên tiếp theo để xét).

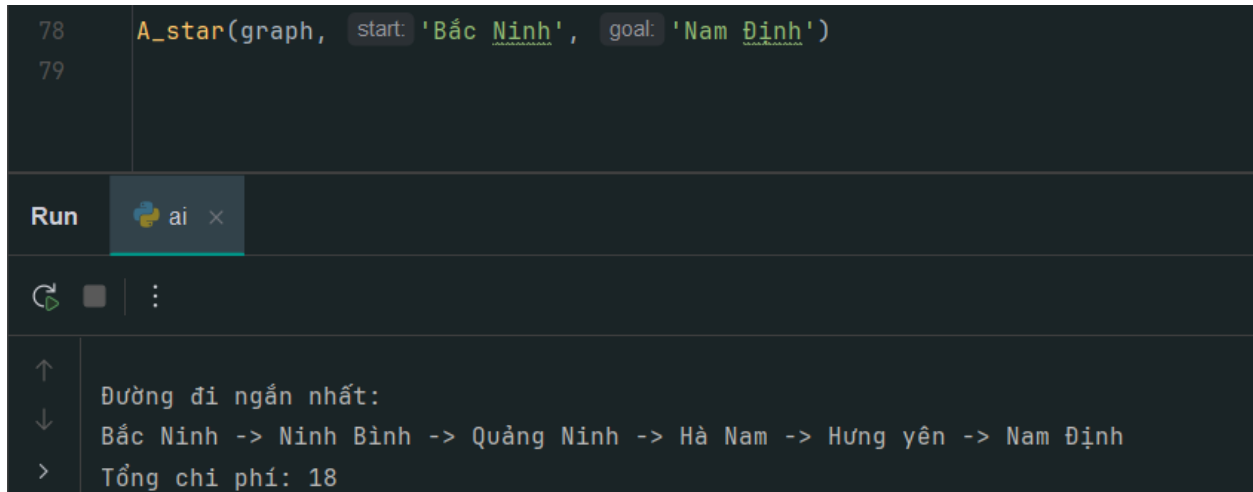
+ Nếu n là một trong các đỉnh đích (có trong goals), thuật toán gọi hàm `print_path_and_cost` để in ra đường đi từ start đến n và chi phí tổng.

+ Nếu n không phải là đích, thuật toán tiếp tục mở rộng các đỉnh lân cận m của n:


- Cập nhật chi phí $g[m]$ từ start đến m.
- Tính giá trị $f(m)$ là tổng chi phí thực tế $g(m)$ và giá trị heuristic $h(m)$.
- Nếu m chưa có trong MO hoặc DONG, thêm m vào MO và cập nhật thông tin của m.
- Nếu m đã có trong MO và có chi phí mới tốt hơn, cập nhật chi phí và cha của m.



+ Lặp lại quá trình cho đến khi tìm được đích hoặc không còn đỉnh nào trong MO.

2.2.4.4. Chạy chương trình tìm đường đi ngắn nhất



```
78 A_star(graph, start: 'Bắc Ninh', goal: 'Nam Định')
79
```

Run  ai ×

↺   ⋮

↑ Đường đi ngắn nhất:
↓ Bắc Ninh -> Ninh Bình -> Quảng Ninh -> Hà Nam -> Hưng yên -> Nam Định
> Tổng chi phí: 18

Hình 2.6 : Chương trình chạy ra đường đi ngắn nhất

KẾT LUẬN

Qua bài tiểu luận, chúng ta đã tìm hiểu rõ hơn về khái niệm không gian trạng thái – một cấu trúc quan trọng trong trí tuệ nhân tạo, đại diện cho tất cả các trạng thái có thể đạt được từ trạng thái ban đầu thông qua các hành động hợp lệ. Đồng thời, các thuật toán tìm kiếm heuristic đã cho thấy vai trò then chốt trong việc tối ưu hoá quá trình tìm kiếm lời giải bằng cách sử dụng các hàm đánh giá heuristic để hướng dẫn việc mở rộng trạng thái, từ đó giúp giảm thiểu chi phí tính toán và thời gian thực hiện. Các thuật toán như A*, Greedy Best-First Search hay thuật toán IDA* không chỉ giúp cải thiện hiệu suất tìm kiếm mà còn mở ra nhiều hướng nghiên cứu và ứng dụng trong các lĩnh vực như robot học, lập lịch, trò chơi và điều khiển tự động. Tuy nhiên, hiệu quả của các thuật toán này vẫn phụ thuộc rất lớn vào việc thiết kế các hàm heuristic phù hợp – vừa chính xác, vừa tính toán nhanh. Tóm lại, việc hiểu và vận dụng tốt các thuật toán tìm kiếm heuristic trong không gian trạng thái không chỉ giúp giải quyết hiệu quả các bài toán trong AI mà còn góp phần nâng cao năng lực tư duy giải quyết vấn đề một cách logic và có hệ thống.

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 nơ-ron phương pháp và ứng dụng, NXB Giáo dục, 2000.
- [4] "Trí tuệ nhân tạo – Cơ sở và ứng dụng" – TS. Nguyễn Minh Thuyết (Nhiều trường đại học dùng làm giáo trình)
- [5] "Trí tuệ nhân tạo" – PGS.TS. Nguyễn Đức Nghĩa (Đại học Bách Khoa TP.HCM)

Bảng đánh giá kết quả

STT	Họ và tên	Mức độ hoàn thành
1	Đinh Hải Anh	20%
2	Nguyễn Minh Ngọc	20%
3	Hà Thành	20%
4	Nguyễn Đức Trung	40%