TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI: ÁP DỤNG THUẬT TOÁN TÌM KIẾM THEO CHIỀU RỘNG VÀ THUẬT TOÁN BEST FIRST SEACH VÀO BÀI TOÁN XẾP HÌNH 8 MẢNH GHÉP.**

**Sinh viên thực hiện : NGUYỄN CẢNH VINH**

**NGUYỄN THỊ NGỌC ÁNH**

**Giảng viên hướng dẫn : NGUYỄN THỊ THANH TÂN**

**Ngành : CÔNG NGHỆ THÔNG TIN**

**Chuyên ngành : CÔNG NGHỆ PHẦN MỀM**

**Lớp : D13CNPM3**

**Khóa : 2018-2023**

***Hà Nội, tháng 10 năm 2020***

# PHIẾU CHẤM ĐIỂM

Sinh viên thực hiện:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Nội dung thực hiện** | **Điểm** | **Chữ ký** |
| 1 | **Nguyễn Cảnh Vinh** | Xác định yêu cầu, thiết kế chương trình, hoàn thành báo cáo và code … |  |  |
| 2 | **Nguyễn Thị Ngọc Ánh** | Xác định yêu cầu và thiết kế chương trình, làm báo cáo, code… |  |  |

Giảng viên chấm điểm:

|  |  |  |
| --- | --- | --- |
| Họ và tên | Chữ ký | Ghi chú |
| Giảng viên chấm 1: |  |  |
| Giảng viên chấm 2: |  |  |

# ĐỀ CƯƠNG

## Mô tả tóm tắt đề tài

Báo cáo có nội dung “Áp dụng thuật toán tìm kiếm theo chiều rộng và thuật toán Best-First Search vào bài toán 8 mảnh”.

Chương đầu của báo cáo nghiên cứu tổng quan về trí tuệ nhân tạo.

Chương hai, trình bày về 2 giải thuật Best-first search và Breadth-First Search.

Chương ba, áp dụng thuật toán vào bài toán 8 mảnh.

Trong ngành khoa học máy tính, một giải thuật tìm kiếm là một thuật toán lấy đầu vào là một bài toán và trả về kết quả là một lời giải cho bài toán đó, thường là sau khi cân nhắc giữa một loạt các lời giải có thể. Tập hợp tất cả các lời giải có thể cho bài toán được gọi là không gian tìm kiếm. Có những thuật toán tìm kiếm “sơ đẳng” không có thông tin, đây là những phương pháp đơn giản và trực quan, trong khi đó các thuật toán tìm kiếm có thông tin sử dụng hàm đánh giá heuristic giúp ta giảm đáng kể thời gian cần thiết cho việc tìm kiếm lời giải. Để áp dụng được các giải thuật tìm kiếm, ta cần chuyển không gian tìm kiếm về dạng đồ thị. Với dạng đồ thị ta sẽ nắm bắt những mối liên hệ, những ảnh hưởng giữa các trạng thái của bài toán một cách nhanh chóng và ngắn gọn. Trong phạm vi bài báo cáo, chúng em xin trình bày hai thuật tìm kiếm toán cơ bản và tiêu biểu với lý thuyết đồ thị đó là: Best-First Search và Breadth-First Search.

## Nội dung thực hiện

-CHƯƠNG 1: TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO

1.Trí tuệ nhân tạo là gì

2. Ứng dụng trí tuệ nhân tạo

3.Phân lại trí tuệ nhân tạo

4. Một số ứng dụng AI điển hình

5. Hạn chế của AI

-CHƯƠNG 2: GIẢI THUẬ TÌM KIẾM BEST-FIRST SEARCH VÀ BREADTH- FIRST SEARCH

1. Các chiến lược tìm kiếm cơ bản trong AI

2. Giải thuật tìm kiếm

-CHƯƠNG 3: BÀI TOÁN 8 MẢNH

1.Mô tả bài toán

2. Hướng dẫn chơi game

3. Ứng dụng thuật toán vào game

4. Các chức năng chính

-CHƯƠNG 4: ĐÁNH GIÁ

1. Vấn đề khó khăn

2. Phương hướng phát triển chương trình

## Kết quả cần đạt được

-Giao diện người dùng và máy được thiết kế một cách khoa học, thân thiện người sử dụng, có tính thống nhất về phương pháp làm việc, cách trình bày.

-Các chức năng trong hệ thống phải hoạt động đúng, chính xác.

# MỤC LỤC

[PHIẾU CHẤM ĐIỂM .](#_Toc53317514)

[MỤC LỤC](#_Toc53317515)

[LỜI MỞ ĐẦU 1](#_Toc53317516)

[CHƯƠNG 1: TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO 2](#_Toc53317517)

[1.1. Trí tuệ nhân tạo là gì 2](#_Toc53317518)

[1.2. Ứng dụng trí tuệ nhân tạo 2](#_Toc53317519)

[1.3. Phân loại trí tuệ nhân tạo 2](#_Toc53317520)

[1.4. Một số ứng dụng AI điển hình 4](#_Toc53317521)

[1.5. Hạn chế của AI 5](#_Toc53317522)

[CHƯƠNG 2: GIẢI THUẬT TÌM KIẾM BEST- FIRST SEARCH VÀ BREADTH- FIRST SEARCH 7](#_Toc53317523)

[2.1. Các chiến lược tìm kiếm cơ bản trong AI 7](#_Toc53317524)

[2.2. Giải thuật tìm kiếm ưu tiên tối ưu Best-First Search 7](#_Toc53317525)

[a. Giải thuật Heuristic 7](#_Toc53317526)

[b. Thuật toán Best-First Search 8](#_Toc53317527)

[c. Thuật toán Breadth- First Search 11](#_Toc53317528)

[CHƯƠNG 3: BÀI TOÁN 8 MẢNH GHÉP 15](#_Toc53317529)

[3.1. Mô tả bài toán 15](#_Toc53317530)

[3.2. Hướng dẫn chơi game 15](#_Toc53317531)

[3.3. Ứng dụng thuật toán vào game ghép tranh 8 số 16](#_Toc53317532)

[3.4. Các chức năng chính 16](#_Toc53317533)

[a. Giao diện chính 16](#_Toc53317534)

[b. Chức năng random hình ảnh trên chương trình 16](#_Toc53317535)

[c. Chức năng đếm bước đi 17](#_Toc53317536)

[d. Chức năng chơi lại 18](#_Toc53317537)

[e. Chức năng tạm dừng 19](#_Toc53317538)

[f. Chức năng giải BFS 19](#_Toc53317539)

[g. Chức năng giải tối ưu 19](#_Toc53317540)

[h. Chức năng thoát 20](#_Toc53317541)

[i. Chức năng đi tới và đi lui 21](#_Toc53317542)

[j. Chức năng tính thời gian 21](#_Toc53317543)

[CHƯƠNG 4: ĐÁNH GIÁ 23](#_Toc53317544)

[4.1. Vấn đề khó khăn 23](#_Toc53317545)

[4.2. Phương hướng phát triển chương trình 23](#_Toc53317546)

[KẾT LUẬN 24](#_Toc53317547)

[TÀI LIỆU THAM KHẢO 25](#_Toc53317548)

# 

# LỜI MỞ ĐẦU

Trí tuệ nhân tạo đã và đang làm thay đổi cuộc sống của chúng ta, với sự phát triển mạnh mẽ của việc áp dụng các nghiên cứu về trí tuệ nhân tạo áp dụng cho cuộc sống. Tất cả các ngành như: Quân đội, y tế, giáo dục, kinh tế thương mại, tài chính,…Đều có thể áp dụng trí tuệ nhân một cách rộng rãi, Việc áp dụng trí tuệ nhân tạo để giải quyết các vấn đề trong xã hội và việc phát triển kinh tế đang được nhà nước khuyến khích và đầu tư rất lớn.

Trên thế giới cũng như Việt Nam, CNTT có ảnh hưởng rất mạnh mẽ đến sự phát triển của đất nước và thế giới đặc biệt là trí tuệ nhân tạo. Nó chở thành một yếu tố không thể thiếu và có tính quyết định đến sự thành công hay thất bại của nhiều ngành ở nước ta, CNTT đang phát triển với tốc độ khá mạnh mẽ và được ứng dụng rộng rãi trong tất cả các lĩnh vực, đặc biệt là trong công tác ứng dụng công nghệ vào trong cuộc sống.

Trong thời đại công nghệ 4.0 đang ngày càng phát triển, ứng dụng trí tuệ nhân tạo vào đời sống được xem là một trong những xu thế phát triển mạnh mẽ hiện nay. Việc tự động hóa các hành vi thông minh đang trở thành một cuộc cách mạng trong ngành CNTT.

Vì vậy chúng em đã thực hiện đồ án trí tuệ nhân tạo **“áp dụng thuật toán “Breadth First Search” và “Best First Search” để làm ra trò chơi ghép tranh 8 ô số”**, phục vụ cho việc chứng minh áp dụng trí tuệ nhân tạo mang lại lợi ích tối ưu về không gian và thời gian cho con người.

Với sự hướng dẫn, động viên tận tình của cô **Nguyễn Thị Thanh Tân**, chúng em đã hiểu được cách thức hoạt động của thuật toán và hoàn thành bài báo cáo đồ án này. Do chưa có nhiều kinh nghiệm nghiên cứu, thực hành nên chúng em cũng không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm và góp ý của thầy để đề tài của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

# CHƯƠNG 1: TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO

## Trí tuệ nhân tạo là gì

**AI- Artificial Intelligence có thể được định nghĩa như một ngành của khoa học máy tính liên quan đến việc tự động hóa các hành vi thông minh.**

AI là một bộ phận của khoa học máy tính và do đó nó phải được đặt trên những nguyên lý lý thuyết vững chắc, có khả năng ứng dụng được của lĩnh vực này.

AI, trí tuệ nhân tạo là sự mô phỏng các quá trình hoạt động trí tuệ của con người, bao gồm quá trình học tập (thu thập thông tin và các quy tắc để sử dụng thông tin), lập luận (sử dụng các quy tắc để đạt được kết luận gần đúng), và tự sửa lỗi.

## Ứng dụng trí tuệ nhân tạo

Một số ứng dụng sử dụng trí tuệ nhân tạo AI bao gồm hệ chuyên gia (expert system), các ứng dụng nhận diện giọng nói và các ứng dụng machine vision.

Trí tuệ nhân tạo, AI được đặt tên bởi nhà khoa học máy tính người Mỹ, John McCarthy vào năm 1956 tại Hội nghị Dartmouth. Ngày nay thuật ngữ AI được bổ sung thêm cả tự động hóa quy trình bằng Robot và Robot thực tế.

Thời gian gần đây nổi lên thuật ngữ mới có tên Big Data hay gia tăng tốc độ, kích thước và thu thập các dữ liệu doanh nghiệp đa dạng. AI có thể thực hiện các tác vụ như xác định các mẫu trong dữ liệu hiệu quả hơn con người, cho phép các doanh nghiệp hiểu rõ hơn về dữ liệu của mình.

## Phân loại trí tuệ nhân tạo

Trí tuệ nhân tạo được phân loại theo nhiều cách khác nhau, dưới đây là 2 ví dụ điển hình về phân loại trí tuệ nhân tạo.

Đầu tiên là phân loại trí tuệ nhân tạo AI theo hệ thống bao gồm trí tuệ nhân tạo mạnh hoặc yếu. Trí tuệ nhân tạo yếu hay còn được gọi là Narrow AI, là hệ thống trí tuệ nhân tạo được thiết kế và đào tạo cho các tác vụ cụ thể. Các trợ lý ảo chẳng hạn như Siri của Apple là một dạng trí tuệ nhân tạo yếu.

Trí tuệ nhân tạo mạnh còn được gọi là Artificial General Intelligence hay trí tuệ nhân tạo tổng hợp, là hệ thống AI được trang bị khả năng nhận thức tổng quát của con người để khi thực hiện các tác vụ không quen thuộc, nó đủ thông minh để tìm ra các giải pháp. Phép thử Turing được phát triển bởi nhà toán học Alan Turing vào năm 1950 là phương pháp được sử dụng để xác thực xem một máy tính có thể có những suy nghĩ giống con người hay không, mặc dù phương pháp này gây nhiều tranh cãi.

Ví dụ thứ 2 là từ Arend Hintze, một trợ lý giáo sư sinh học hợp nhất và khoa học máy tính và kỹ thuật tại Đại học bang Michigan. Ông phân loại trí tuệ nhân tạo AI thành 4 loại, từ loại hệ thống AI hiện nay đến các hệ thống cảm giác, mà chưa tồn tại.

**Loại 1: Máy phản ứng**. Một ví dụ điển hình là Deep Blue, chương trình cờ vua của IBM đánh bại Garry Kasparov vào những năm 1990. Deep Blue có thể xác định các phần trên bảng cờ vua và dự đoán, nhưng nó không có bộ nhớ và không thể sử dụng những kinh nghiêm trong quá khứ để thông báo cho con người trong tương lai. Nó phân tích các động thái của chính mình và đối thủ, và chọn một động thái chiến lược nhất. Deep Blue và AlphaGO của Google được thiết kế cho mục đích hẹp và không thể dễ dàng áp dụng cho tình huống khác.

**Loại 2: Bộ nhớ hạn chế**. Các hệ thống AI này có thể sử dụng các kinh nghiệm trong quá khứ để đưa ra các quyết định trong tương lai. Một số chức năng ra quyết định trong các loại xe tự lái được thiết kế dựa trên cách này. Các quan sát được sử dụng để thông báo cho các hành động xảy ra trong tương lai không xa, chẳng hạn như một chiếc xe đã thay đổi làn đường. Những quan sát này không được lưu trữ vĩnh viễn.

**Loại 3: Lý thuyết.** Đây là một thuật ngữ tâm lý. Thuật ngữ này đề cập đến việc hiểu được rằng con người có niềm tin, ham muốn và ý định của chính họ ảnh hưởng đến quyết định của họ. Tuy nhiên loại AI này chưa tồn tại.

**Loại 4: Tự nhận thức.** Với phân loại này, hệ thống AI có ý thức về bản thân. Các máy có ý thức tự giác hiểu được trạng thái hiện tại của chúng và có thể sử dụng thông tin để suy ra những gì người khác đang cảm nhận. Loại AI này vẫn chưa tồn tại.

- Tự động hóa là quá trình tạo ra hệ thống hoặc chức năng xử lý tự động. Ví dụ như tự động hóa quy trình bằng Robot để thực hiện các tác vụ khối lượng lớn mà con người thường xuyên lặp lại. Tự động hóa quy trình bằng Robot khác với tự động hóa công nghệ thông tin ở chỗ nó có thể thích nghi với các hoàn cảnh thay đổi khác nhau.

- Machine Learning là khoa học để một máy tính thực hiện các hành động mà không cần lập trình. Deep Learning là tập hợp con của Machine Learning, với các thuật ngữ khá đơn giản, có thể được coi là tự động hóa của các phân tích tiên đoán.

Có 3 thuật toán Machine Learning, bao gồm: **Supervised Learning** (học có giám sát), trong đó tập hợp dữ liệu được gắn nhãn sao cho các mẫu được phát hiện và sử dụng để gắn nhãn các tập dữ liệu mới. Tiếp theo là **Unsupervised Learning** (học không có giám sát), trong đó tập hợp dữ liệu không được dán nhãn và được sắp xếp theo phân loại tương đồng hoặc khác biệt. Và cuối cùng là **Reinforcement Learning** (học tăng cường), trong đó tập hợp dữ liệu không được dán nhãn, nhưng sau khi thực hiện một hành động hoặc một số hành động, hệ thống AI được phản hồi.

- Machine Vision (hệ thống nhận dạng và điều khiển dựa trên hình ảnh) là khoa học làm cho máy tính có thể nhìn thấy được. Machine Vision thu thập và phân tích thị giác thông tin bằng cách sử dụng một camera, chuyển đổi từ kỹ thuật số và xử lý tín hiệu số. Nó thường được so sánh với thị giác của con người, nhưng machine vision không bị ràng buộc bởi sinh học và có thể được lập trình để xem qua các bức tường. Ví dụ điển hình là machine vision được sử dụng trong một loạt các ứng dụng từ nhận dạng chữ ký để phân tích hình ảnh y tế. Machine Vision, tập trung xử lý hình ảnh dựa trên máy.

- Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) là quá trình xử lý ngôn ngữ máy tính của con người chứ không phải máy tính. Một trong những ví dụ nổi tiếng nhất phải kể đến phát hiện thư rác, xem dòng chủ đề và văn bản của một email và quyết định xem đó có phải là thư rác hay không. NLP xử lý các tác vụ bao gồm dịch văn bản, phân tích tình cảm và nhận dạng giọng nói.

- Nhận dạng mẫu (Pattern recognition) là một phần của Machine Learning, trong đó tập trung vào việc xác định các mẫu trong dữ liệu.

-Robotics là một lĩnh vực kỹ thuật tập trung vào việc thiết kế và sản xuất robot. Robot thường được sử dụng để thực hiện các tác vụ khó khăn cho con người, và được sử dụng trong dây chuyền lắp ráp để sản xuất ô tô hoặc do NASA vận chuyển các vật thể to lớn trong không gian. Gần đây hơn, các nhà nghiên cứu đang sử dụng machine learning để xây dựng các robot có thể tương tác trong môi trường xã hội.

## Một số ứng dụng AI điển hình

Dưới đây là một số ứng dụng điển hình của trí tuệ nhân tạo:

- **AI trong lĩnh vực chăm sóc sức khỏe**: AI góp phần cải thiện tình trạng sức khỏe bệnh nhân, đồng thời giảm các chi phí điều trị. Các công ty đang áp dụng Machine Learning để chẩn đoán nhanh hơn và tốt hơn con người. Một trong những công nghệ chăm sóc sức khỏe tốt nhất phải kể đến IBM Watson, có khả năng hiểu được các ngôn ngữ tự nhiên và có khả năng phản hồi các câu hỏi được yêu cầu. Hệ thống này khai thác dữ liệu bệnh nhân và các nguồn dữ liệu sẵn có khác để tạo ra giả thuyết. Sau đó, nó sẽ trình bày một lược đồ điểm tin cậy. Các ứng dụng khác của AI bao gồm chatbot, chương trình máy tính trực tuyến để trả lời các câu hỏi và hỗ trợ khách hàng, sắp xếp các cuộc hẹn hoặc trợ giúp bệnh nhân thông qua quá trình thanh toán và các trợ lý y tế ảo cung cấp phản hồi y tế cơ bản.

**- AI trong lĩnh vực kinh doanh:**Tự động hóa quy trình bằng Robot được áp dụng cho các tác vụ mà con người thực hiện lặp đi lặp lại. Thuật toán Machine Learning được tích hợp trên các nền tảng phân tích và CRM để khám phá các thông tin về cách phục vụ khách hàng tốt hơn. Chatbots được tích hợp trên các trang web để cung cấp dịch vụ ngay lập tức cho khách hàng.

**- AI trong lĩnh vực giáo dục:**AI có thể tự động phân loại, giúp người làm giáo dục có thể tiết kiệm một khoảng thời gian đáng kể. AI có thể đánh giá sinh viên và thích ứng với nhu cầu của họ. Đồng thời AI có thể hỗ trợ thêm cho sinh viên làm thêm công việc gia sư, đảm bảo rằng họ có thể đi đúng hướng.

**- AI trong lĩnh vực tài chính**: AI áp dụng cho các ứng dụng tài chính cá nhân như Mint hay Turbo Tax, tăng cường các định chế tài chính. Một số ứng dụng khác như IBM Watson được áp dụng AI này cho các giao dịch mua bán nhà.

- **AI trong lĩnh vực pháp luật**: Quá trình khám phá, chọn lọc thông qua các tài liệu trong luật pháp thường áp đảo đối với con người. Tự động hóa quá trình này giúp tiết kiệm thời gian và quá trình làm việc hiệu quả hơn. Startups cũng đang xây dựng các trợ lý ảo cho máy tính hỏi và trả lời các câu hỏi được lập trình. Hơn nữa, chúng có thể sàng lọc các câu hỏi được lập trình để trả lời bằng cách kiểm tra phân loại.

**- AI trong lĩnh vực sản xuất**: Đây là lĩnh vực đi đầu trong việc kết hợp robot vào luồng công việc. Robot công nghiệp được sử dụng để thực hiện các nhiệm vụ đơn lẻ và đã được tách ra khỏi con người.

## Hạn chế của AI

**Trí tuệ nhân tạo mang lại rất nhiều giá trị cho cuộc sống loài người, nhưng cũng tiềm ẩn những nguy cơ.**

Rất nhiều chuyên gia lo lắng rằng khi trí tuệ nhân tạo đạt tới 1 ngưỡng tiến hóa nào đó thì đó cũng là thời điểm loài người bị tận diệt. Rất nhiều các bộ phim đã khai thác đề tài này với nhiều góc nhìn, nhưng qua đó đều muốn cảnh báo loài người về mối nguy đặc biệt này.

Dự báo cho rằng từ 5 đến 10 năm nữa, ngành khoa học này sẽ phát triển lên tới đỉnh cao. Hãy cùng chờ đợi những thành tựu mới nhất của loài người về lĩnh vực này

# CHƯƠNG 2: GIẢI THUẬT TÌM KIẾM BEST- FIRST SEARCH VÀ BREADTH- FIRST SEARCH

## Các chiến lược tìm kiếm cơ bản trong AI

**Tìm kiếm mù:** Chiến lược tìm kiếm mù là kỹ thuật tìm kiếm mà trong đó chúng ta không có hiểu biết gì về các đối tượng để có hướng dẫn tìm kiếm mà chỉ đơn thuần xem xét các đối tượng theo một hệ thống nào đó để phát hiện ra đối tượng cần tìm.

**Tìm kiếm kinh nghiệm**: Chiến lược tìm kiếm kinh nghiệm (tìm kiếm heuristic) là kỹ thuật tìm kiếm dựa vào kinh nghiệm và sự hiểu biết của chúng ta về vấn đề cần giải quyết để xây dựng nên hàm đánh giá hướng dẫn sự tìm kiếm.

**Tìm kiếm có đối thủ:** Thủ tục tìm kiếm cho các bài toán liên quan đến 2 đối tượng (đối thủ) có mục tiêu đối nghịch nhau (xung đột với nhau). Phương pháp tìm kiếm có đối thủ được áp dụng phổ biến trong các trò chơi (games).

## Giải thuật tìm kiếm

### Giải thuật Heuristic

**Heuristics**: là các phỏng đoán, ước chừng dựa trên kinh nghiệm, trực giác. Heuristics có thể được thể hiện dưới dạng luật hoặc hàm số.

Thuật giải Heuristic là một sự mở rộng khái niệm thuật toán. Nó 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 của con người.

Có nhiều phương pháp để xây dựng một thuật giải Heuristic, trong đó người ta thường dựa vào một số nguyên lý cơ sở như sau:

***Nguyên lý vét cạn thông minh :***

Trong một bài toán tìm kiếm nào đó, khi không gian tìm kiếm lớn, ta thường tìm cách giới hạn lại không gian tìm kiếm 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 (Greedy):***

Lấy tiêu chuẩn tối ưu (trên phạm vi toàn cục) của bài toán để làm tiêu chuẩn chọn lựa hành động cho phạm vi cục bộ của từng bước (hay từng giai đoạn) 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 một lời giải tốt.

***Hàm Heuristic:***

Trong việc xây dựng các thuật giải Heuristic, người ta thường dùng các *hàm Heuristic*. Ðó là các hàm đánh giá thô, giá trị của hàm phụ thuộc vào trạng thái hiện tại của bài toán tại mỗi bước giải. Nhờ giá trị này, ta có thể chọn được cách hành động tương đối hợp lý trong từng bước của thuật giải.

### Thuật toán Best-First Search

Tìm kiếm theo lựa chọn tốt nhất ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): *Best-first search*) là một [thuật toán tìm kiếm](https://vi.wikipedia.org/wiki/Gi%E1%BA%A3i_thu%E1%BA%ADt_t%C3%ACm_ki%E1%BA%BFm) tối ưu hóa [tìm kiếm theo chiều rộng](https://vi.wikipedia.org/wiki/T%C3%ACm_ki%E1%BA%BFm_theo_chi%E1%BB%81u_r%E1%BB%99ng) bằng cách mở rộng nút hứa hẹn nhất được chọn theo một quy tắc nào đó. Để có được hiệu quả về thời gian chạy cho việc chọn ra ứng cử viên tốt nhất cho việc mở rộng, người ta thường dùng một [hàng đợi ưu tiên](https://vi.wikipedia.org/w/index.php?title=H%C3%A0ng_%C4%91%E1%BB%A3i_%C6%B0u_ti%C3%AAn&action=edit&redlink=1) để cài đặt cấu trúc dữ liệu lưu trữ các lựa chọn hiện hành.

Tìm kiếm theo bề rộng được hướng dẫn bởi hàm đánh giá hay còn gọi là hàm Heuristic. Nằm trong nhóm thuật toán tìm kiếm kinh nghiệm.Hàm đánh giá được tính theo 2 phương thức: đếm số ô sai vị trí so với trạng thái người dùng mong muốn và tính theo khoảng cách Mahattan. Trong trò chơi này hàm đánh giá (Heuristic) được tính theo đếm số ô sai vị trí so với trạng thái đích mong muốn.

Ưu điểm của tìm kiếm theo chiều sâu là không phải quan tâm đến sự mở rộng của các nhánh. Ưu điểm của tìm kiếm theo chiều rộng là không bị sa vào các đường dẫn bế tắc (các nhánh cụt). Tìm kiếm ưu tiên tối ưu sẽ kết hợp 2 phương phát trên cho phép ta đi theo một con đường duy nhất tại một thời điểm, nhưng đồng thời vẫn “quan sát” được những hướng khác. Nếu con đường đang đi “ có vẻ” không triển vọng bằng những con đường ta đang “quan sát” ta sẽ chuyển sang đi theo một trong số các con đường này.

Một cách cụ thể, tại mỗi bước của tìm kiếm BFS, ta chọn đi theo trạng thái có khả năng cao nhất trong số các trạng thái được xét cho đến thời điểm đó. Khác với leo đồi dốc đứng 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 như tìm kiếm leo đồi dốc đứng), nhưng ta sẽ không bị lẩn quẩn trong các nhánh này vì nếu càng đi sau vào một hướng mà ta phát hiện ra rằng hướng này càng đi thì càng tệ, 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 hương tốt nhất trong số những hướng chưa đi.

**-Mô tả thuật toán: :** Cũng tương tự như thuật toán Breadth First Search. Nhưng Best First Search khác ở chổ: Đỉnh được chọn để phát triển là đỉnh tốt nhất được xác định bởi hàm đánh giá (đỉnh có giá trị hàm đánh giá nhỏ nhất). Có nghĩa là mỗi lần thêm vào hàng đợi danh sách các trạng thái sẽ được sắp xếp theo thứ tự tăng dần dựa theo hàm đánh giá, hàm đánh giá càng thấp thì tương đương số ô bị sai vị trí là ít nhất. Từ đó ta chọn được trạng thái tốt nhất để đi tiếp.

**-Ưu điểm:**

+ Ưu điểm của thuật toán này duyệt qua ít đỉnh hơn vì có hàm đánh giá ở trong nên khi duyệt việc sắp xếp hàm đánh giá sẽ giúp tìm tới đỉnh cuối cùng nhanh hơn và ít tốn thời gian hơn.

+Dễ cài đặt.

+Nếu số đỉnh là hữu hạn, thuật toán chắc chắn tìm ra kết quả.

+Có xét thêm hàm Heuristic nên duyệt ít trạng thái hơn, Breadth First Search => nhanh hơn, ít tốn không gian hơn.

**-Nhược điểm:**

+Chiếm thời gian và không gian lớn, nếu đỉnh duyệt quá lớn.

**-Thuật toán:**

+OPEN: Chứa tập các trạng thái đã được sinh ra nhưng chưa được xét đến, có cấu trúc 1 hàng đợi ưu tiên (priority queue).

• Các phần tử được sắp xếp theo độ tốt (giá trị của hàm heuristic).

• Phần tử có độ ưu tiên cao nhất là phần tử tốt nhất.

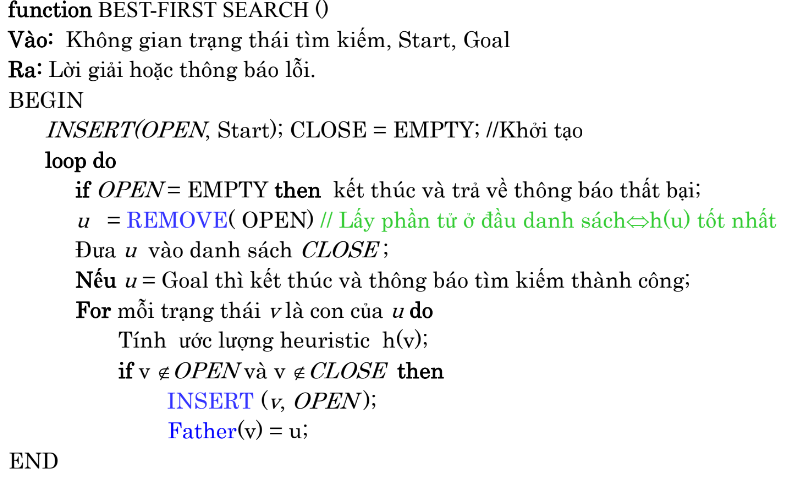
+CLOSE: Danh sách các trạng thái đã xét.

+Hàm REMOVE(): lấy phần tử ở đầu danh sách OPEN.

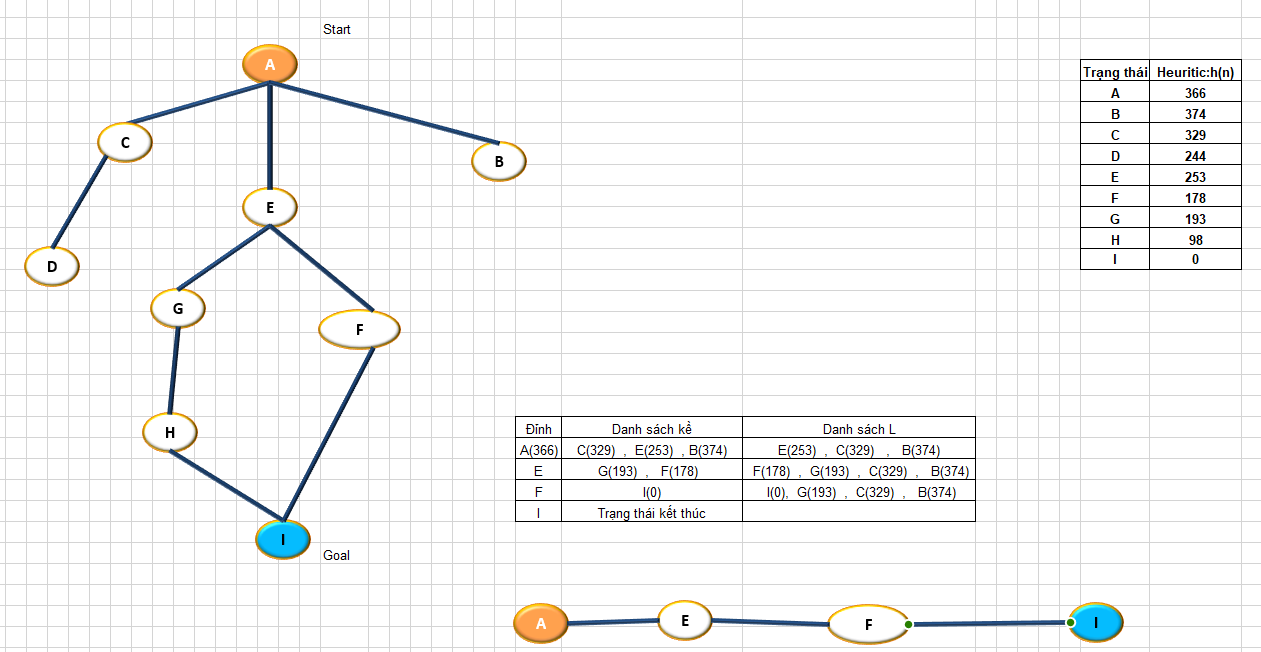
+Hàm INSERT(): Chèn phần tử vào danh sách OPEN theo thứ tự độ ưu tiên giảm dần (hàm heuristic h tăng dần).

+Father (x): Cha của x.

-**Cài đặt thuật toán:**



**VD:**



Hình 2.2.b: Ví dụ thuật tóan Best -First Search

Ta thấy đường đi cuối cùng vẫn là A->E->F->I như là ví dụ . Nhưng số trạng thái mà thuật toán phải duyệt chỉ có 4 đỉnh là đã tìm ra được đường đi.

Do trong danh sách L (Queue hàng đợi) lúc này đã đưa được trạng thái tốt nhất về đầu danh sách để lấy ra duyệt đi tiếp.

### Thuật toán Breadth- First Search

Đây là thuật toán tìm đường đi từ đỉnh xuất phát đến đỉnh kết thúc bằng các duyệt theo chiều rộng. Đây là thuật toán nằm trong nhóm thuật toán tìm kiếm mù, thuật toán không quan tâm đến trọng số trên đường đi mà chỉ duyệt theo những đỉnh kề liên tiếp nó. Xuất phát tử một đỉnh và đi tới các đỉnh kề nó, tiếp tục cho đến khi không còn đỉnh nào để đi. Trong quá trình đi đến đỉnh kề, tiến hành lưu lại đỉnh kề để khi đi ngược lại từ đỉnh kết thúc đến đỉnh xuất phát ta có được đường đi ngắn nhất.

**-Mô tả thuật toán: :** Các đỉnh đã được xét thì không thể xét lại lần 2 nữa. Cơ chế lưu lại đỉnh kề sẽ là lưu các đỉnh kề thành 1 danh sách và lấy từ từ danh sách các đỉnh kề ra để xét, khi mà một đỉnh kề của một đỉnh đang xét được thêm vào danh sách thì nó sẽ được thêm vào cuối của danh sách hay còn gọi là cơ chế Queue(hàng đợi). Cơ chết này có nghĩa khi thêm một phần tử thì phần tử đó sẽ được thêm ở cuối danh sách hàng đợi, còn lấy phần tử ra thì sẽ lấy ra ở đầu danh sách hàng đợi. Đó chính là điểm nổi bật để phân biệt thuật toán này với các thuật toán khác.

**-Ưu điểm:**

+Dễ cài đặt.

+Nếu số đỉnh là hữu hạn thuật toán chắc chắn tìm ra kết quả.

**-Nhược điểm:**

+Mang tính chất vét cạn, không nên áp dụng nếu duyệt nếu số đỉnh quá lớn.

+Mang tính chất mù quáng, duyệt tất cả đỉnh không chú ý đến thông tin trong các đỉnh để duyệt hiệu quả, dẫn đến duyệt các đỉnh không cần thiết.

+Chiếm thời gian và không gian bộ nhớ khi số duyệt đỉnh nhiều.

**-Thuật toán:**

**Input:** cây đồ thị G= (V,E) với đỉnh gốc là s0 ( trạng thái đầu) Tập đích Goals.

**Output:** một đường đi p từ n0 đến 1 đỉnh f trong tập Goals. Thuật toán sử dụng một cấu trúc dữ liệu là hàng đợi (Queue) để lưu trữ thông tin trung gian trong quá trình tìm kiếm (ở đây dễ hiểu là các đỉnh kế tiếp đợi được duyệt). Tương tự với tìm kiếm theo chiều sâu, các bước cho giải thuật tìm kiếm theo chiều rộng như sau :

**Bước 1:** Khởi tạo

- Các đỉnh đều ở trạng thái chưa đánh dấu, trừ đỉnh xuất phát s là đã đánh dấu.

- Một hàng đợi Q (tổ chức dạng hàng đợi Queue), ban đầu chỉ có một phần tử là s. Hàng đợi dùng để chứa các đỉnh sẽ được duyệt theo thứ tự ưu tiên chiều rộng.

**Bước 2:** Lặp lại các bước sau cho đến khi hàng đợi rỗng:

- Nếu hàng đợi rỗng, không thấy đỉnh đích, thông báo “ không tìm thấy” , dừng.

- Hàng đợi không rỗng, lấy u ra khỏi hàng đợi , thông báo thăm u (bắt đầu duyệt đỉnh u, nếu là lần duyệt đầu thì u ở đây là s).

- Kiếm tra u có phải đỉnh đích t không

* Nếu đúng trả về u, dừng vòng lặp, sang bước 3
* Nếu sai tiếp tục chương trình.

- Xét tất cả các đỉnh v kề với u mà chưa được đánh dấu, với mỗi đỉnh v đó:

* Đánh dấu v
* Ghi nhận đường đi từ v đến u
* Đấy v vào hàng đợi (v sẽ chờ được duyệt tại những bước sau).
* **Bước 3:** Truy ngược lại đường đi.

– OPEN: Cấu trúc lưu trữ hàng đợi QUEUE.

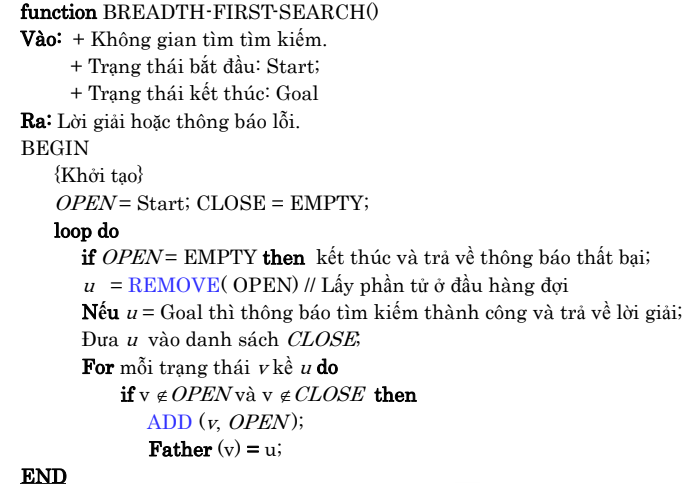
– CLOSE: Danh sách các trạng thái đã xét.

– Father (x): Cha của x;

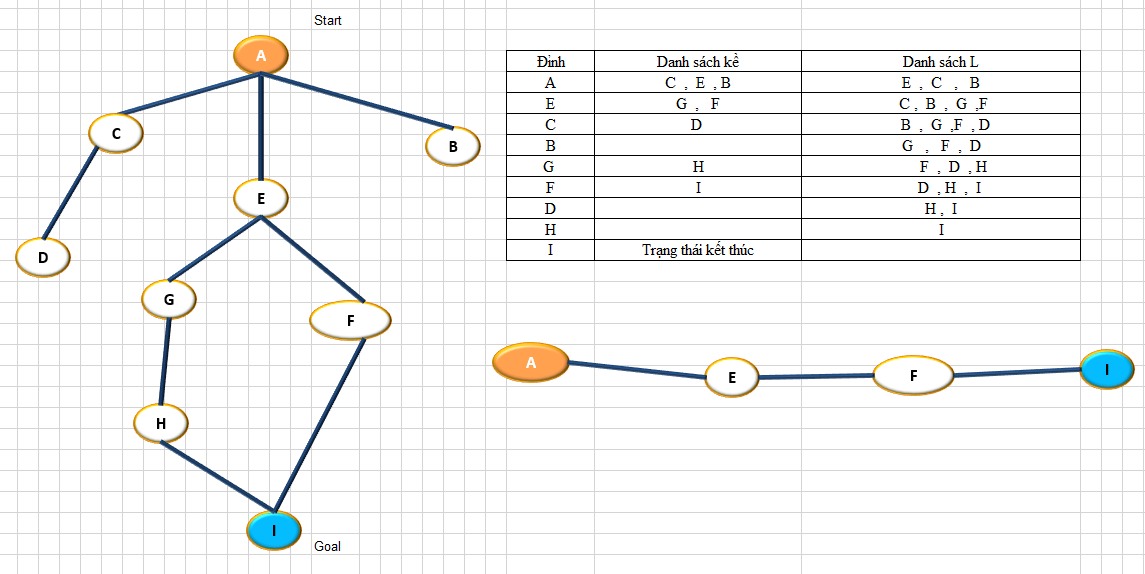
– REMOVE (): lấy phần tử ở đầu hàng đợi OPEN.

– ADD(): Đưa phần vào cuối hàng đợi OPEN.

**-Cài đặt thuật toán:**

****

**VD:**



Hình 2.2.c: Ví dụ về thuật toán Breadth- First Search

Trong ví dụ trên, các trạng thái trong danh sách kề sẽ được thêm lần lượt vào cuối trong danh sách L(Queue hàng đợi), mỗi lần đỉnh duyệt sẽ lấy ra trạng thái đầu tiên trong danh sách L này để duyệt tiếp theo. Cứ tiếp tục như vậy cho đến khi đi đến đỉnh kết thúc cần tìm hoặc cho đến khi Danh sách L (Queue hàng đợi ) đã rỗng.

# CHƯƠNG 3: BÀI TOÁN 8 MẢNH GHÉP

## Mô tả bài toán

Vị trí của các hình trong trò chơi sẽ nằm ngẫu nhiên trộn lẫn trong 9 ô, trong đó có 1 ô đen để người dùng dịch chuyển đi từng bước. Mỗi lần di chuyển người dùng chỉ có thể đi 1 bước theo chiều qua trái, qua phải, đi lên hoặc đi xuống để ghép thành 1 hình hoàn chỉnh theo hình mẫu đã cho theo đó. Người dùng **không** được đi chéo. Trong quá trình chơi có thể có trường hợp người dùng không thể đi đến trạng thái hoàn chỉnh của hình. Vì vậy chúng ta áp dụng trí tuệ nhân tạo vào trong trò chơi này, và cụ thể là chúng em đã áp dụng 2 thuật toán Breadth-First-Search và Best-First-Search trong game này để đưa người chơi đi đến trạng thái hoàn chỉnh (giải ra được đường đi đến trạng thái hoàn chỉnh). Vậy bài toán thực tế ở đây là tìm ra đường đi đến trạng thái hoàn thành của game ghép tranh 8 ô số, cũng như đưa một trạng thái ngẫu nhiên chưa hoàn chỉnh của bức tranh về một bức tranh hoàn chỉnh áp dụng 2 thuật toán trong trí tuệ nhân tạo.

## Hướng dẫn chơi game

Khi người dùng khởi động chương trình lên. Màn hình sẽ hiển thị bên táy phải là hình gốc (hình hoàn chỉnh sau ghi ghép đúng bức tranh), bên tay trái là 9 ô số chứa từng phần nhỏ được cắt ra từ bức tranh hoàn chỉnh, 9 ô số này sẽ có 1 ô là màu đen trống để người chơi di chuyển các ô lân cận, 9 ô này sẽ được trộn lẫn ngẫu nhiên không hoàn chỉnh thành 1 bức tranh. Nhiệm vụ người chơi là di chuyển từng ô 1 lên, xuống, trái, phải và không được đi chéo dựa theo ô trống đen, sao cho ghép thành 1 bức tranh hoàn chỉnh như hình bên tay phải.

Khi người chơi bắt đầu chơi, thời gian sẽ được tính, và số bước đi sẽ được đếm nhằm thống kê thời gian và số bước người chơi đã đi đến kết quả cuối cùng. Người chơi có thể tạm ngưng trong lúc chơi, nhưng khi chọn nút tạm ngưng màn hình 9 ô số ngẫu nhiên này sẽ bị tắt đi, phòng trường hợp người chơi xem trước đường đi.Người chơi có thể chọn nút chơi lại để bắt đầu chơi lại từ đầu, lúc này 9 ô số sẽ được trộn ngẫu nhiên lại một lần nữa và thời gian cũng như bước đi sẽ được cài đặt về 0 như trạng thái ban đầu.Nếu người chơi không thể đi đến ghép thành một bức tranh hoàn chỉnh thì người chơi có thể chọn nút giải theo BFS hay giải Tối Ưu trên màn hình, sau khi chọn thuật toán giải số bước đi sẽ được hiển thị lên, thời gian và số trạng thái đã duyệt qua cũng được hiển thị lên, từ đây người chơi click vào nút đi tới để bắt đầu xem kết quả đi từng bước của thuật toán cho đến khi hoàn chỉnh bức tranh. Người dùng cũng có thể đi lui để xem lại bước đi trước đó.

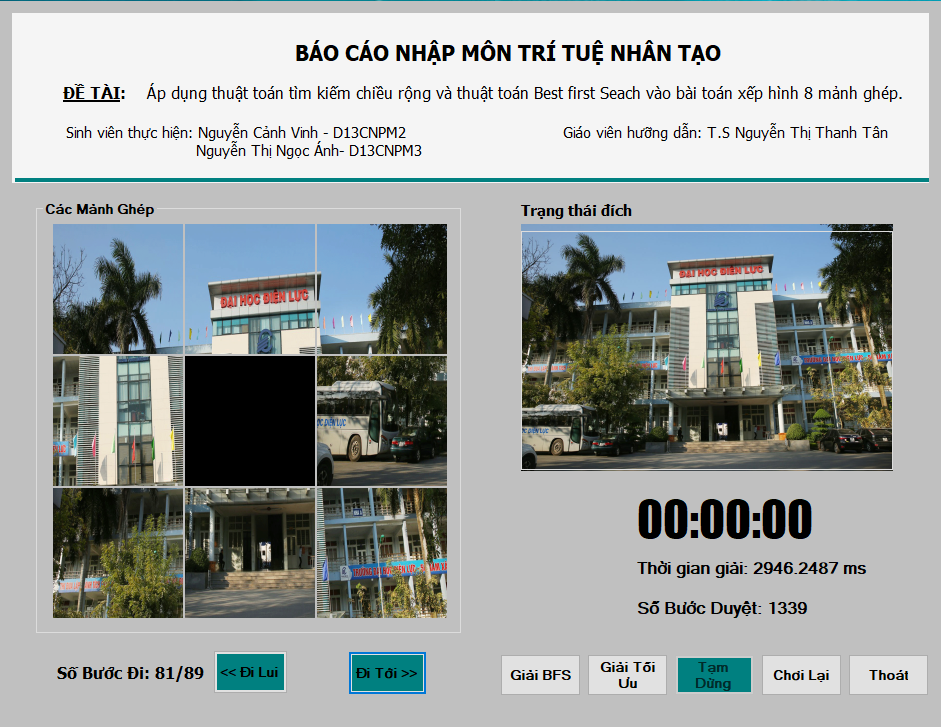
Người dùng có thể thoát khỏi chương trình.

## Ứng dụng thuật toán vào game ghép tranh 8 số

Áp dụng 2 thuật toán như đã nói trên vào game 8 ô số. Lúc này mỗi trạng thái hay mỗi đỉnh mà thuật toán duyệt qua chính là một danh sách các số từ 1 đến 9, và trong đó có 1 ô số đánh dấu là ô đen trống để cho người chơi di chuyển. Mỗi lần thuật toán duyệt qua một trạng thái, sẽ đưa vào trong hàng đợi, như vậy ta sẽ có danh sách chứa những danh sách. Kết quả đường đi tìm được sẽ trả về danh sách của những trạng thái mà nó đã tìm ra(cũng là một danh sách các số từ 1 đến 9).

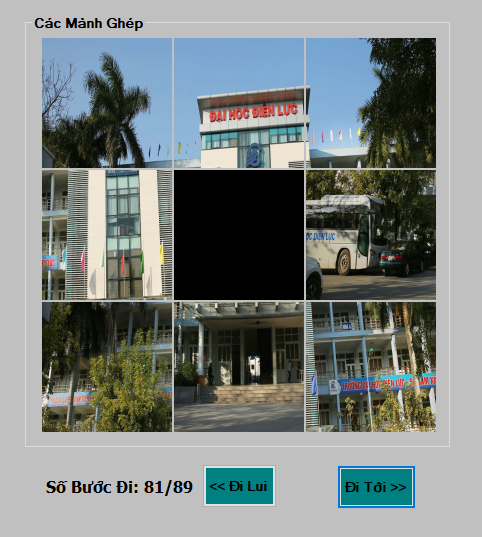
## Các chức năng chính

### Giao diện chính



### Chức năng random hình ảnh trên chương trình

- Khi bước vào trò chơi thì hình ảnh sẽ được sắp xếp một cách ngẫu nhiên.



### Chức năng đếm bước đi

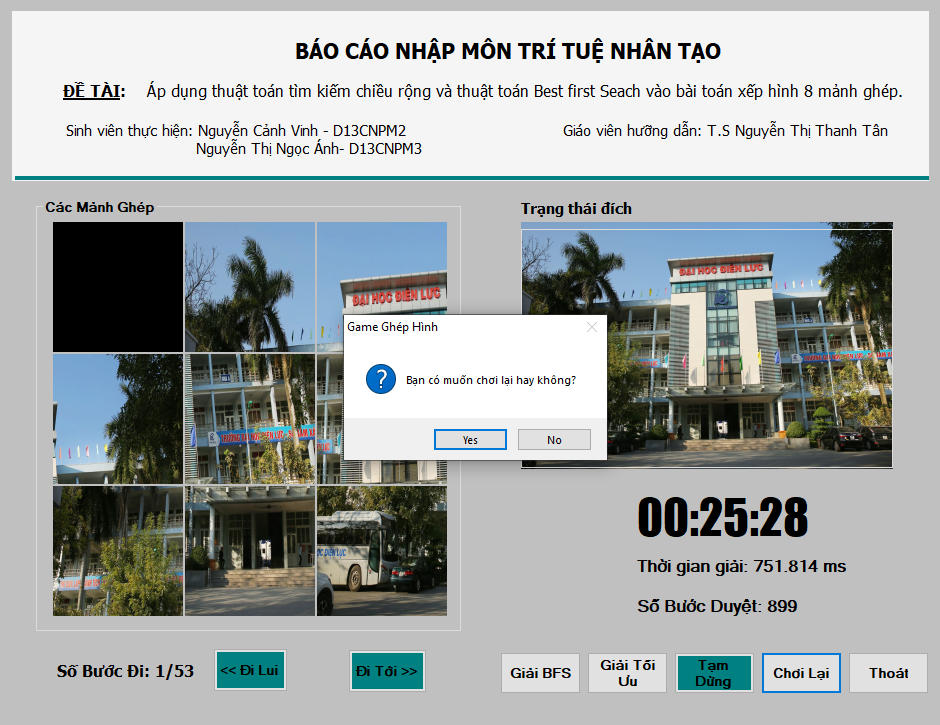


-Khi chúng ta di chuyển các trang thái để sắp xếp hình ảnh thì mỗi lần di chuyển sẽ được tính là một bước.

-Chức năng này giúp đánh giá được bạn đã sử dụng bao nhiêu trạng thái để ghép thành công tấp hình này. Hay gọi cách khác là bạn đã sử dụng bao nhiêu trang thái để hoàn thành trò chơi này.

-Ngoài đánh giá chức năng này còn có thể giúp bạn ghi nhận lại số bước đi để lần tới bạn có thể cải thiện hơn số trạng thái mình đã dùng để chiến thắng trò chơi. Một khi bạn dùng càng ít trạng thái để chiến thắng thì bạn sẽ đánh giá được khả năng chơi trò chơi của mình như thế nào.

### Chức năng chơi lại

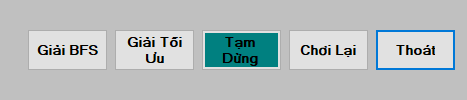


-Khi người chơi muốn chơi gặp khó khăn trong việc di chuyển các ô trên màn hình trò chơi hoặc là người chơi đã tìm ra cách để chiến thắng trò chơi mà tốt ít thời gian nhất cũng như là tốn ít số bươc đi để đến chiến thắng nhất thì người chơi có thể chơi lại từ đầu

-Khi bạn cảm thấy chơi hoài mà không chiến thắng thì bạn cũng có thể bấm chơi lại để bắt đầu lại từ đầu. Khi đó sẽ có một thông báo hỏi bạn có muốn chơi lại từ đầu hay không.

-Khi bạn đã chắc chắn là mình chơi lại từ đầu thì màn hình chơi sẽ random lại các trạng thái một cách ngẫu nhiên.

### Chức năng tạm dừng



-Khi mới vào màn hình chơi game thì nút tạm dừng sẽ bị ẩn. Tới khi bạn chơi và bước đi cũng như thời gian được tính thì bạn mới có thể tạm dừng được.

-Chức năng này giúp bạn đang chơi thì bạn sẽ có thể tạm dừng.

-Để đảm báo tính hấp dẫn và thú vị cho trò chơi thì khi bạn bấm tạm dừng thì hình ảnh trên màn hình của bạn sẽ mất và chỉ còn để lại tấm hình mẫu bên phải mà thôi. Còn các trạng thái của bạn đang sắp xếp dở thì sẽ biến mất. Khiến cho bạn không thể tạm dừng để suy nghĩ các bước đi của mình được.

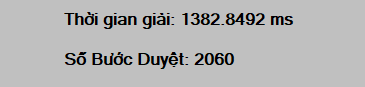
-Khi đó button tạm dừng sẽ chuyển thành tiếp tục để người chơi chọn chơi tiếp.

### Chức năng giải BFS

Click vào button giải BFS thì người dừng phải đợi một chút để máy có thể tự giải. Tới khi nào phần đếm bước đi hiển thị số bước:



Thì khi đó thuật toán đã được giải và sẽ hiển thị ra màn hình cả thời gian và số bước duyệt để chọn ra được những trạng thái cuối cùng để người chơi có thể đi đến chiến thắng.

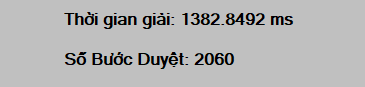


### Chức năng giải tối ưu

Khi bạn click vào button giải tối ưu thì người dùng sẽ chờ một chút. Tới khi mà phần đến bước đi hiển thị ra số bước đi thì khi đó trò chơi mới tự được máy giải xong.



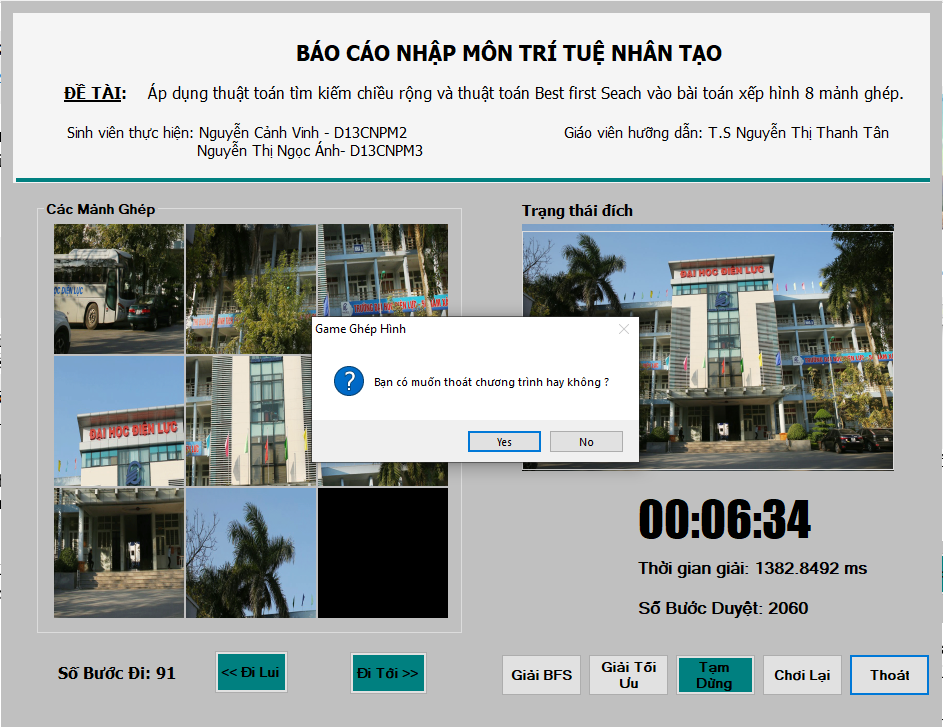
Chức năng này cũng sẽ hiển thi ra các thời gian và số trạng thái đã được duyệt qua để đi đến trạng thái kết thúc.

****

Việc hiển thị số bước để duyệt hay còn gọi là số trạng thái để đi đến trạng thái cuối cùng, sẽ giúp người dùng so sánh được hai phương pháp giải với nhau.

### Chức năng thoát

Khi muốn tắt chương trình thì người dùng sẽ có chức năng thoát. Khí bạn click vào thì sẽ hiển thị một thông báo xác nhận là bạn có muốn thoát hay không.



Người dùng chọn có thì chương trình sẽ thoát còn chọn không thì chương trình vẫn sẽ tiếp tục.

### Chức năng đi tới và đi lui



Khi bạn đã chọn chức năng giải BFS hoặc giải tối ưu chương trình đã thực thi xong thì bạn sẽ chọn chức năng đi tới và đi lui để đi từng bước tới trạng thái kết thúc.

Chức năng này giúp người chơi có thể hiểu được các bước giải của máy thông qua việc đi từng bước từng bước.

Khi đã đi hết tất cả các bước thì bạn sẽ tới trạng thái kêt thúc.

### Chức năng tính thời gian



Khi bắt đầu vào trò chơi thì thời gian vẫn sẽ chưa được tính, cho đến khi bạn bắt đầu di chuyển trạng thái đầu tiên của một ô trên khung thì thời gian sẽ bắt đầu tính. Hiểu nôm na sẽ là bạn di chuyển một ô thì thời gian sẽ bắt đầu tính kể cả khi sau đó bạn không di chuyển thêm một ô nào khác thì thời gian vẫn sẽ tính.

Chức năng tính thời gian cũng sẽ là một chức năng đánh giá khả năng của bạn tương tự như đếm bước đi của bạn vậy. Bạn hoàn thành trong thời gian càng ngắn thì chứng tỏ bạn rất thông minh và chơi game này rất tốt.

Thời gian sẽ dừng khi trạng thái bạn hoàn thành sẽ là trạng thái kết thúc. Hay nói cách khác là bạn đã xếp đúng hình thì thời gian sẽ dừng lại.

# CHƯƠNG 4: ĐÁNH GIÁ

## 4.1. Vấn đề khó khăn

Trường hợp nếu là trạng thái đầu truyền vào là một mảng random thì nếu gặp random ra số vị trí ô sai nhiều thì cả 2 thuật toán Breadth-Frist-Search và Best First Search đều mất quá nhiều thời gian để tìm ra được đường đi. (Có thể em khắc phục bằng cách tạo ra 5 trường hợp tương đương 5 testcase có sẵn, 5 trường hợp này đều là các trạng thái mà 2 thuật toán có thể tìm ra được đường đi trong thời gian không quá lâu từ 0.01-> dưới 10 giây. Và cho trạng thái đầu vào chạy random trong 5 testcase này.)

Nếu người chơi bấm nút tạm ngừng, sau đó bấm nút chơi lại thì chương trình sẽ bị lỗi.

## 4.2. Phương hướng phát triển chương trình

Đối với trường hợp có quá nhiều ô bị sai thì thuật toán Best First Search hay Breadth First Search vẫn tìm ra đường đi quá lâu, cần áp dụng các thuật toán tối ưu hơn như A\*, Nhánh Cận, AKT để có sự quan sát độ tối ưu, thời gian …v…v… cao hơn.

Nên in đường đi là các trạng thái đã duyệt qua lên màn hình giao diện dưới dạng mảng 2 chiều, thay vì là đã in dưới Console.

Đề xuất khi bấm nút giải thì label thời gian phải bắt đầu đếm chứ không phải giải xong rồi mới hiển thị thời gian cuối cùng lên lable.

Đề xuất khi bấm nút giải tự động các picture box phải tự di chuyển đi đến trạng thái cuối cùng hoàn chỉnh mà không cần người chơi phải click nút “đi tới” để xem.

Mở rộng trò chơi lên 16 ô, 25 ô.

# KẾT LUẬN

Thông qua việc tìm kiếm và nghiên cứu đề tài này giúp chúng em có cái nhìn toàn diện hơn trong việc ứng dụng trí tuệ nhân tạo vào giải quyết vấn đề thực tế. Đây là bài toàn cổ điển trong trí tuệ nhân tạo vào giải quyết vấn đề thực tế. Đây là bài toán cổ điển trong trí tuệ nhân tạo cho các thuậ toán mô hình hóa liên quan đén tìm kiếm có tri thức bổ sung. Đề tài đã được nhiều người nghiên cứu giải quyết, nhưng cho đến nay vẫn chưa có cách giải quyết tối ưu cho tất cả không gian trạng thái trò chơi vì kích thước tăng không gian trạng thái sẽ tăng lên rất nhanh.

Trong quá trình tìm hiểu và làm bài, nhóm em đã cố gắng hoàn thiện bài làm và hoàn thiện mình hơn về mặt kiến thức. Nhưng cũng không thể tránh khỏi những sai sót, mong thầy cô và các bạn góp ý để bài của nhóm hoàn thiện hơn.

# TÀI LIỆU THAM KHẢO

[1]. Nhập môn Trí tuệ nhân tạo, TS Ngô Hữu Phúc (Học viện kĩ thuật Quân Sự).

[2]. Trí tuệ nhân tạo Artificial Intelligence, Phạm Thọ Hoàn, Phạm Thị Anh Lê, Trường Đại học Sư Phạm Hà Nội.

[3]. Website: <https://www.brainkart.com/article/Best-First-Search--Concept,-Algorithm,-Implementation,-Advantages,-Disadvantages_8881//>.

[4].Website:<https://en.wikibooks.org/wiki/Artificial_Intelligence/Search/Exhaustive_search/Breadth-first_search>.

[5]. Website: <https://www.geeksforgeeks.org/c-sharp-random-next-method/>