C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

**BÁO CÁO PROJECT 1**

**1612272- TRẦN NHẬT HUY**

**1612282-TRẦN ĐÌNH KHẢI**

**NHẬP MÔN CƠ SỞ TRÍ TUỆ NHÂN TẠO**

**Contents**

[**1. THÔNG TIN NHÓM VÀ BÀI TOÁN 2**](#_Toc526850849)

[**1.1. Thông tin nhóm 2**](#_Toc526850850)

[**1.2. Tóm tắt bài toán: Tìm đường đi: 2**](#_Toc526850851)

[**2. PHÂN CÔNG CÔNG VIỆC 3**](#_Toc526850852)

[**3. SƠ ĐỒ BIỂU DIỄN HỆ THỐNG 4**](#_Toc526850853)

[**4. MÔ TẢ CẤU TRÚC DỮ LIỆU 4**](#_Toc526850854)

[**5. MÔ TẢ THUẬT TOÁN 4**](#_Toc526850855)

[**6. CÁC BỘ TEST 4**](#_Toc526850856)

[**7. MỨC ĐỘ HOÀN THÀNH VÀ ĐÁNH GIÁ 5**](#_Toc526850857)

# THÔNG TIN NHÓM VÀ BÀI TOÁN

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **HỌ VÀ TÊN** | **EMAIL** | **SỐ ĐIỆN THOẠI** |
| 1612272 | TRẦN NHẬT HUY | [nhathuy13598@gmail.com](mailto:nhathuy13598@gmail.com) | 0354 878 677 |
| 1612282 | TRẦN ĐÌNH KHẢI | trandinhkhai10.1@gmail.com | 01679969653 |

## Thông tin nhóm

## Tóm tắt bài toán: Tìm đường đi:

Không gian di chuyển được biểu diễn thành ma trận vuông NxN, mỗi ô có một giá trị 0 là ô trống, hoặc 1 là chướng ngại vật

**O**

**y**

**x**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |

Minh hoạ với ma trận 5x5, các ô đen là chướng ngại vật

Các vị trí được xác định bởi (x,y) theo hệ toạ độ ở hình trên bắt đầu từ (0,0). Cho điểm đầu với điểm đích. Sử dụng thuật toán tìm kiếm A\* với h(n) là khoảng cách Euclidean tìm đường đi ngắn nhất <né chướng ngại vật>. Thứ tự mở của ô đang xét:

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 8 |  | 4 |
| 7 | 6 | 5 |

# PHÂN CÔNG CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| **Công việc** | **Người thực hiện** | **Mục tiêu/ sản phẩm** |
| Viết báo cáo | Khải | File báo cáo |
| Vẽ sơ đồ hệ thống | Huy | Ảnh sơ đồ |
| Vẽ cấu trúc dữ liệu | Huy | Ảnh UML |
| Viết mô tả thuật toán | Khải | Văn bản trong phần báo cáo |
| Xây dựng các bộ test | Huy | Các file input và output tương ứng |
| Code thuật toán | Khải, Huy | Source code |

# SƠ ĐỒ BIỂU DIỄN HỆ THỐNG

main():

openfile(filename, mode) -> tuple

A(start:Node,goal:Node,matrix:list)

heuristic(self:Node,g:Node)

subNode(self:Node,goal:Node,data:list)

|  |  |
| --- | --- |
| Tên hàm | Công dụng |
| heuristic(self,g) | Tính khoảng cách Euclid từ 1 đỉnh đến đỉnh goal |
| subNode(self,goal,data) | Mở rộng các node kề cận với node self. Node này chưa được đi qua và không là chướng ngại vật trong mảng các node data |
| A(start:Node,goal:Node,matrix:list) | Thuật toán A\* tìm đường đi từ start đến goal với mảng chứa các đỉnh matrix |
| main() | Hàm main |
| openfile(filename, mode) -> tuple | Hàm mở file để đọc dữ liệu gồm tên file (filename), chế độ đọc (mode). Hàm trả về tuple gồm: (ma trận điểm, điểm đầu, điểm đích) |

# MÔ TẢ CẤU TRÚC DỮ LIỆU

|  |  |
| --- | --- |
|  | Class Node |
| **Attributes** | x: Hoành độ một node |
| y: Tung độ một node |
| g: Chi phí đi |
| f: Chi phí tổng gồm Heuristic + Chi phí đi (g) |
| stt: Số thứ tự mở |
| parent: Node cha |
| **Methods** | \_\_init\_\_(self,x,y,parent)  Hàm khởi tạo với tọa độ x, y và node cha (parent) |
| heuristic(self,goal)  Hàm tính heuristic từ một đỉnh đến đỉnh goal |
| subNode(self,goal,data:list)  Hàm mở rộng các node có thể đến được bằng node self. Khi mở rộng có tính heuristic đến goal và node đó có phải chướng ngại vật bằng mảng các node (data) |
| \_\_lt\_\_(self, other)  Hàm so sánh 2 node để lấy 1 node từ hang đợi ưu tiên |

**Mã giả trong C:**

struct Node{

int x,

int y,

int heuristic,

int parent

};

# MÔ TẢ THUẬT TOÁN

**Mã giả trong C/C++ thuật toán A\***

Node result[size(Ma trận) \* size(Ma trận)];

<priority-queue> A

A.Thêm\_node(Start)

result[Start.x\*size + Start.y] = -1;

loop do{

if (A = rỗng)

then return Thất\_bại;

Node temp = Lấy\_node\_ưu tiên nhất(A);

result[temp.x\*size + temp.y] = temp.parent;

if (temp = Goal)

then return Lời\_giải;

Mở\_rộng(temp,A);

}

**Mã giả mở rộng node**

int check[size(Ma trận) \* size(Ma trận)] = {0}

int step\_X[] = {-1,-1,-1,0,1,1,1,0};

int step\_Y[] = {-1,0,1,1,1,0,-1,-1};

function Mở\_rộng(node, A){

Node temp;

int i = 0;

loop do{

if (node.x + step\_X[i] và node.y + step\_Y[i] thuộc ma trận && Ô đó có check = 0 && Ô đó không phải là chướng ngại vật){

temp.x = node.x + step\_X[i];

temp.y = node.y + step\_Y[i];

temp.heuristic = Euclid(temp,Goal);

temp.parent = node.x\*size(Ma trận) + node.y;

A.Thêm\_node(temp);

Đánh dấu check = 1 là đã mở;

}

i += 1;

}

}

**Mã giả lời giải**

function Lời\_Giải(Goal){

Node temp = Goal;

printf("(%d,%d)", Goal.x, Goal.y);

while (temp.parent != -1){

printf(" <-- ");

printf("(%d,%d)", result[temp.parent].x, result[temp.parent].y);

temp = result[temp.parent];

}

}

# CÁC BỘ TEST

**Bộ dữ liệu 1:**

Điểm start là (0,0)

Điểm goal là (6,6)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Kết quả

11

(0,0) (1,1) (1,2) (2,3) (3,4) (4,3) (5,2) (6,3) (6,4) (6,5) (6,6)

S - - - - - -

- x x - - o -

- o o x - o -

- o o - x o -

- o o x - o -

- o x o o o -

- - - x x x G

**Bộ dữ liệu 2:**

Điểm start là (4,1)

Điểm goal là (0,2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |

Kết quả

5

(4,1) (3,1) (2,2) (1,3) (0,2)

- - G - -

- o o x -

- - x - o

- x o - -

- S o - -

**Bộ dữ liệu 3:**

Điểm start là (3,0)

Điểm goal là (0,0)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

Kết quả

11

(3,0) (4,0) (5,1) (5,2) (5,3) (4,4) (3,4) (2,3) (1,2) (1,1) (0,0)

G o - o - - o o

- x x - - o - o

o o - x - o - o

S o o o x - - -

x o - o x - - -

- x x x - - - -

- - o - - o - -

- o - - - - - o

# MỨC ĐỘ HOÀN THÀNH VÀ ĐÁNH GIÁ

|  |  |  |  |
| --- | --- | --- | --- |
| **Công việc** | **Người thực hiện** | **Mục tiêu/ sản phẩm** | **Đánh giá (%)** |
| Viết báo cáo | Khải | File báo cáo | 100% |
| Vẽ sơ đồ hệ thống | Huy | Ảnh sơ đồ | 100% |
| Vẽ cấu trúc dữ liệu | Huy | Ảnh UML | 100% |
| Viết mô tả thuật toán | Khải | Văn bản trong phần báo cáo | 100% |
| Xây dựng các bộ test | Huy | Các file input và output tương ứng | 100% |
| Code thuật toán | Khải, Huy | Source code | 100% |