ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

CƠ SỞ TRÍ TUỆ NHÂN TẠO

BÁO CÁO ĐỒ ÁN THỰC HÀNH

PROJECT 1 - Robot tìm đường

LỚP CỬ NHÂN TÀI NĂNG

Nhóm thực hiện:

1712152 - Nguyễn Thị Mai Thanh

1712856 - Huỳnh Văn Tú

1712858 - Nguyễn Ngọc Tú

MỤC LỤC

I. Thông tin nhóm:

Họ và tên	MSSV
Nguyễn Thị Mai Thanh	1712152
Huỳnh Văn Tú	1712856
Nguyễn Ngọc Tú	1712858

II. Mức độ hoàn thành:

Mức độ	Mức độ hoàn thành
Mức 1	100%
Mức 2	100%
Mức 3	100%
Mức 4	100%
Mức 5	0% (Chưa cài đặt)

III. Chi tiết thuật toán:

Để tránh trường hợp số thực, khoảng cách dùng trong các thuật toán dưới đây giữa 2 điểm được tính như sau : **AB= |XA - XB| + |YA – YB|**, theo công thức này nếu AB càng nhỏ thì số bước từ A đến B càng ít, nên có thể dùng cho hàm heuristic.

1. Thuật toán BFS (Breadth First Search)

Chi tiết thuật toán: Chèn đỉnh START vào hàng đợi

B1: Lấy ra đỉnh đầu tiên trong hàng đợi và đánh dấu đỉnh này đã được viếng thăm. Sau đó kiểm tra:

- Nếu đỉnh này chính là GOAL, dừng quá trình tìm kiếm và trả về đường đi.
- Nếu không phải thì chèn tất cả các đỉnh kề với đỉnh này mà **chưa được** viếng thăm trước đó.

B2: Nếu hàng đợi là rỗng, thì tất cả các đỉnh có thể đến được đều đã được quan sát => Dừng việc tìm kiếm và trả về "Không tìm thấy đường đi".

B3: Nếu hàng đợi không rỗng thì quay về bước 1.

*2. Thuật toán A**

+ *Mô tả thuật toán:*

A* lưu giữ một tập các đường đi qua đồ thị, bắt dầu từ nút xuất phát. Tập lời giải này được lưu trong hàng đợi ưu tiên. Thứ tự ưu tiên gán cho một đường đi x được quyết định bởi hàm $h(x) = f(x) + g(x)$. Hàm $h(x)$ có giá trị càng thấp thì độ ưu tiên của x càng cao.

+ *Chi tiết thuật toán:*

- Chọn hàm heuristic: **h(x) = f(x) + g(x)**

với f(x) là khoảng cách từ x tới GOAL và g(x) là khoảng cách từ START tới x

- Chèn đỉnh **START** vào hàng đợi

B1: Lấy ra đỉnh đầu tiên trong hàng đợi ưu tiên và đánh dấu đỉnh này đã được viếng thăm. Sau đó kiểm tra:

- Nếu đỉnh này chính là GOAL, dừng quá trình tìm kiếm và trả về kết quả.
- Nếu không phải thì chèn tất cả các đỉnh kề với đỉnh này và **chưa được** viếng thăm trước đó, trong quá trình chèn ta sẽ ưu tiên chèn các đỉnh có heuristic *h* thấp lên trước, nếu bằng nhau ta sẽ ưu tiên điểm là **nước đi xéo** lên trước.

B2: Nếu hàng đợi là rỗng, thì tất cả các đỉnh có thể đến được đều đã được quan sát => dừng việc tìm kiếm và trả về "Không tìm thấy đường đi".
B3: Nếu hàng đợi không rỗng thì quay về bước 1.

3. Thuật toán Greedy Best-First Search

+ *Mô tả thuật toán:*

Thuật toán sẽ sử dụng một hàm đánh giá h(x) là chi phí để đi từ x đến nút đích. Phương pháp này sẽ ưu tiên xét các nút có vẻ gần với nút đích nhất.

+ *Chi tiết thuật toán:* Chèn đỉnh **START** vào hàng đợi

B1: Lấy ra đỉnh đầu tiên trong hàng đợi ưu tiên và đánh dấu đỉnh này đã được viếng thăm. Sau đó kiểm tra:

- Nếu đỉnh này chính là GOAL, dừng quá trình tìm kiếm và trả về kết quả.
- Nếu không phải thì chèn tất cả các đỉnh kề với đỉnh này và **chưa được** viếng thăm trước đó, trong quá trình chèn ta sẽ ưu tiên chèn các đỉnh gần **GOAL** nhất lên trước, nếu bằng nhau ta sẽ ưu tiên điểm là **nước đi xéo** lên trước.

B2: Nếu hàng đợi là rỗng, thì tất cả các đỉnh có thể đến được đều đã được quan sát – dừng việc tìm kiếm và trả về "Không tìm thấy đường đi".
B3: Nếu hàng đợi không rỗng thì quay về bước 1.

IV. Cấu trúc chương trình:

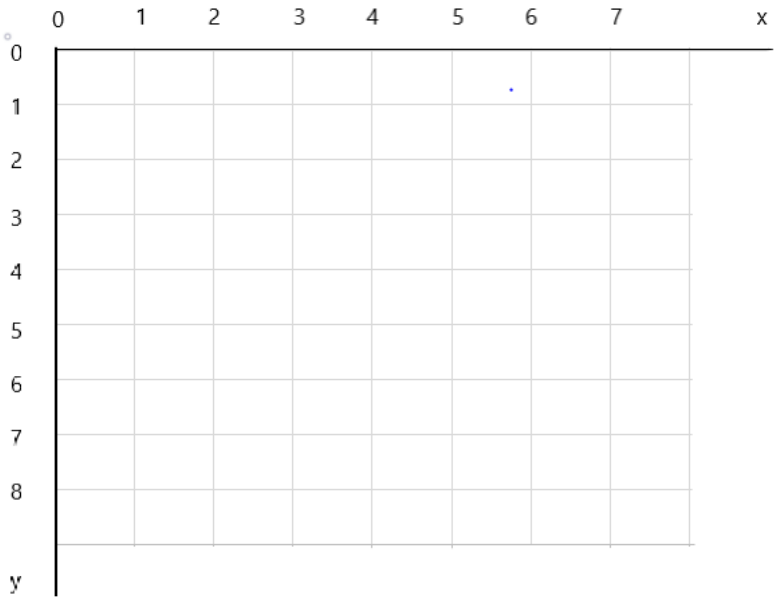
1. Các lớp đối tượng:

Gồm 4 lớp đối tượng:
+ Point - các điểm trong đồ thị với 2 giá trị x, y tương ứng với hoành độ và tung độ
+ Person - người với một giá trị kiểu Point là tọa độ hiện tại của người trên đồ thị
+ Polygon - các đa giác trong không gian với danh sách các đỉnh của đa giác và mỗi đỉnh là 1 Point
+ Map - Không gian của bài toán với các giá trị:
    * topLeft và bottomRigt: có dạng Point - là gốc tọa độ và kích thước giới hạn của không gian bài toán
    * Start, Goal: điểm xuất phát và điểm kết thúc dạng Point
    * Person: vị trí hiện tại của người dạng Person
    * ListPol: có dạng Polygon - là danh sách các đa giác trong không gian
    * Sum_Pixel: là số thực - tổng chi phí đường đii từ S => G
    * Solution: Lộ trình đường đi từ S => G với mỗi phần tử là 1 Point
    * ListPickUp: có dạng Point - Danh sách các điểm đón

2. Giao diện:

- Sử dụng thư viện Pygame - một pixel sẽ có kích thước CONST_SCALE * CONST_SCALE với CONST_SCALE tùy chỉnh.

- Với 2 đỉnh liên tiếp, tìm phương trình đường thẳng qua 2 điểm, sau đó vẽ từng pixel dựa vào phương trình đã tìm được
- Trục không gian được biểu diễn như hình bên dưới:



V. Các ví dụ và trường hợp đặc biệt:

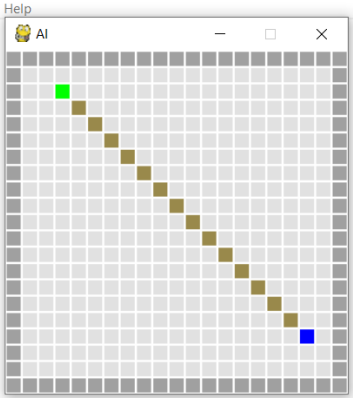
1. Mức 1: Cài đặt 1 thuật toán tìm đường từ S tới G

Sử dụng thuật toán Greedy để tìm đường đi từ S => G.
Chi phí lộ trình được tính theo công thức: số pixel đi dọc/ngang + 1.5 * số pixel đi xéo

Ví dụ 1: Không gian không có vật cản

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (18,17)
(3,2)->(4,3)->(5,4)->(6,5)->(7,6)->(8,7)->(9,8)->(10,9)->(11,10)->(12,11)
->(13,12)->(14,13)->(15,14)->(16,15)->(17,16)->(18,17)
Độ dài đường đi:  23.5
```

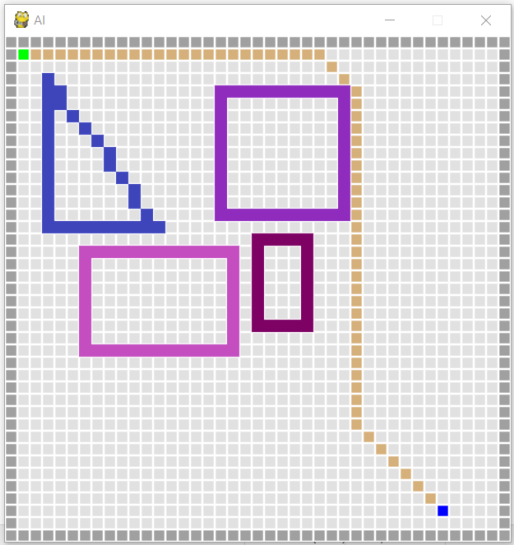
```
File Edit Format View Help
20,20
3,2,18,17
0
0
0,1
```



Ví dụ 2: Không gian có vật cản và có đường đi từ S => G

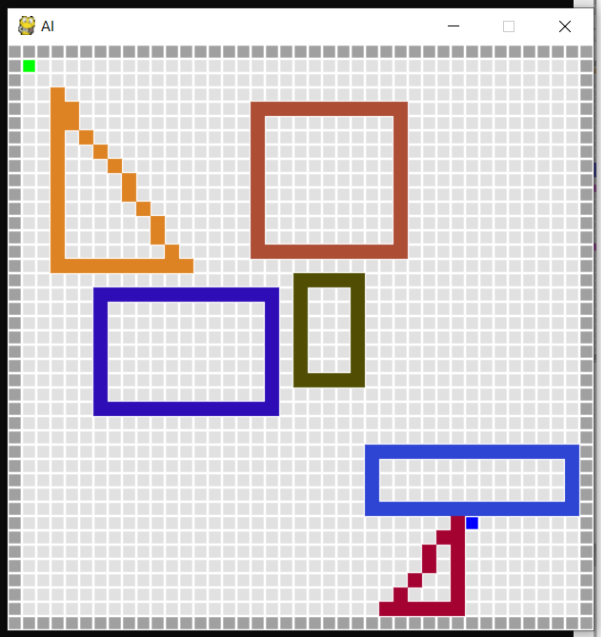
```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (35,38)
(1,1)->(2,1)->(3,1)->(4,1)->(5,1)->(6,1)->(7,1)->(8,1)->(9,1)->(10,1)->(1
1,1)->(12,1)->(13,1)->(14,1)->(15,1)->(16,1)->(17,1)->(18,1)->(19,1)->(20
,1)->(21,1)->(22,1)->(23,1)->(24,1)->(25,1)->(26,2)->(27,3)->(28,4)->(28,
5)->(28,6)->(28,7)->(28,8)->(28,9)->(28,10)->(28,11)->(28,12)->(28,13)->(
28,14)->(28,15)->(28,16)->(28,17)->(28,18)->(28,19)->(28,20)->(28,21)->(2
8,22)->(28,23)->(28,24)->(28,25)->(28,26)->(28,27)->(28,28)->(28,29)->(28
,30)->(28,31)->(29,32)->(30,33)->(31,34)->(32,35)->(33,36)->(34,37)->(35,
38)
Độ dài đường đi:  67.0
```

```
File Edit Format View Help
40,40
1,1,35,38
4
6,17,6,25,18,25,18,17
17,4,17,14,27,14,27,4
20,16,24,16,24,23,20,23
3,3,3,15,12,15
0
0,1
```



Ví dụ 3: Không gian có vật cản và không tồn tại đường đi từ S => G

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (32,33)

Không tìm thấy đường đi tới đích!!!
```




```
File Edit Format View Help
40,40
1,1,32,33
6
6,17,6,25,18,25,18,17
17,4,17,14,27,14,27,4
20,16,24,16,24,23,20,23
3,3,3,15,12,15
25,32,39,32,39,28,25,28
31,39,26,39,31,33
0
0,1
```

2. Mức 2: Cài đặt thành công 3 thuật toán để tìm đường đi từ S tới G

Cài đặt 3 thuật toán tìm đường đi từ S => G là Greedy, BFS và A*

Chi phí đường đi được tính như Level 1

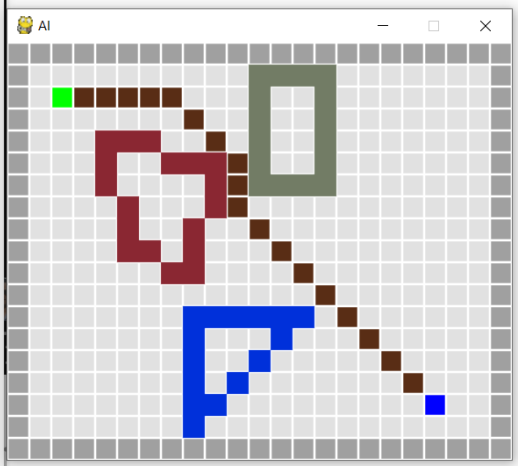
a. Greedy:

Có đường đi từ S => G:

Ví dụ 1:

```
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (19,16)
(2,2)->(3,2)->(4,2)->(5,2)->(6,2)->(7,2)->(8,3)->(9,4)->(10,5)->(10,6)->
(10,7)->(11,8)->(12,9)->(13,10)->(14,11)->(15,12)->(16,13)->(17,14)->(18
,15)->(19,16)
Độ dài đường đi:  26.0
```

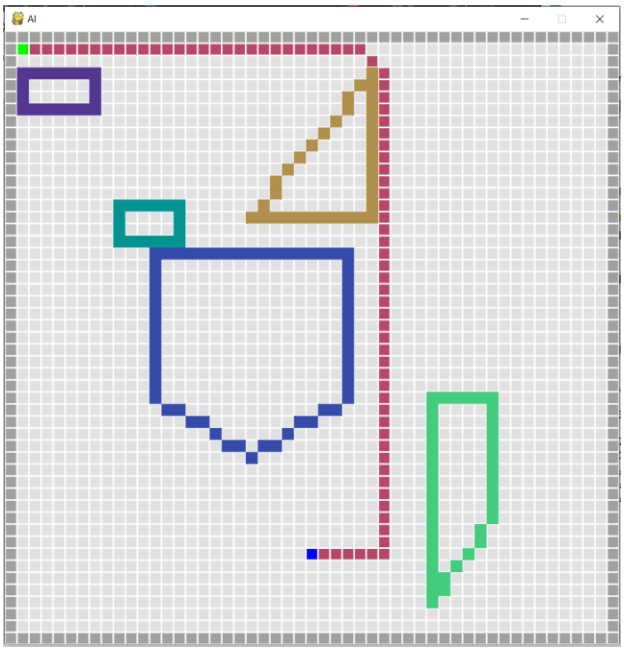
```
File  Edit  Format  View  Help
22,18
2,2,19,16
3
4,4,5,9,8,10,9,5
8,12,8,17,13,12
11,1,11,6,14,6,14,1
0
0,1
```



Ví dụ 2:

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (25,43)
(1,1)->(2,1)->(3,1)->(4,1)->(5,1)->(6,1)->(7,1)->(8,1)->(9,1)->(10,1)->(1
1,1)->(12,1)->(13,1)->(14,1)->(15,1)->(16,1)->(17,1)->(18,1)->(19,1)->(20
,1)->(21,1)->(22,1)->(23,1)->(24,1)->(25,1)->(26,1)->(27,1)->(28,1)->(29,
1)->(30,2)->(31,3)->(31,4)->(31,5)->(31,6)->(31,7)->(31,8)->(31,9)->(31,1
0)->(31,11)->(31,12)->(31,13)->(31,14)->(31,15)->(31,16)->(31,17)->(31,18
)->(31,19)->(31,20)->(31,21)->(31,22)->(31,23)->(31,24)->(31,25)->(31,26)
->(31,27)->(31,28)->(31,29)->(31,30)->(31,31)->(31,32)->(31,33)->(31,34)-
>(31,35)->(31,36)->(31,37)->(31,38)->(31,39)->(31,40)->(31,41)->(31,42)->
(31,43)->(30,43)->(29,43)->(28,43)->(27,43)->(26,43)->(25,43)
Độ dài đường đi:  78.0
```

```
File  Edit  Format  View  Help
50,50
1,1,25,43
5
1,3,1,6,7,6,7,3
9,14,14,14,14,17,9,17
40,30,40,40,35,47,35,40,35,30
12,18,12,30,20,35,28,30,28,18
20,15,30,15,30,3
0
0,1
```

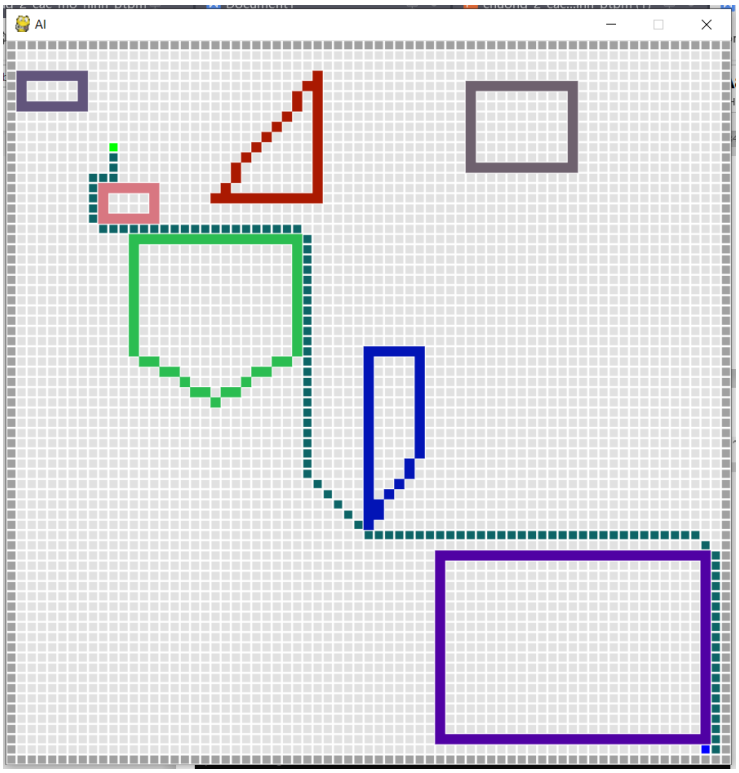


Ví dụ 3:

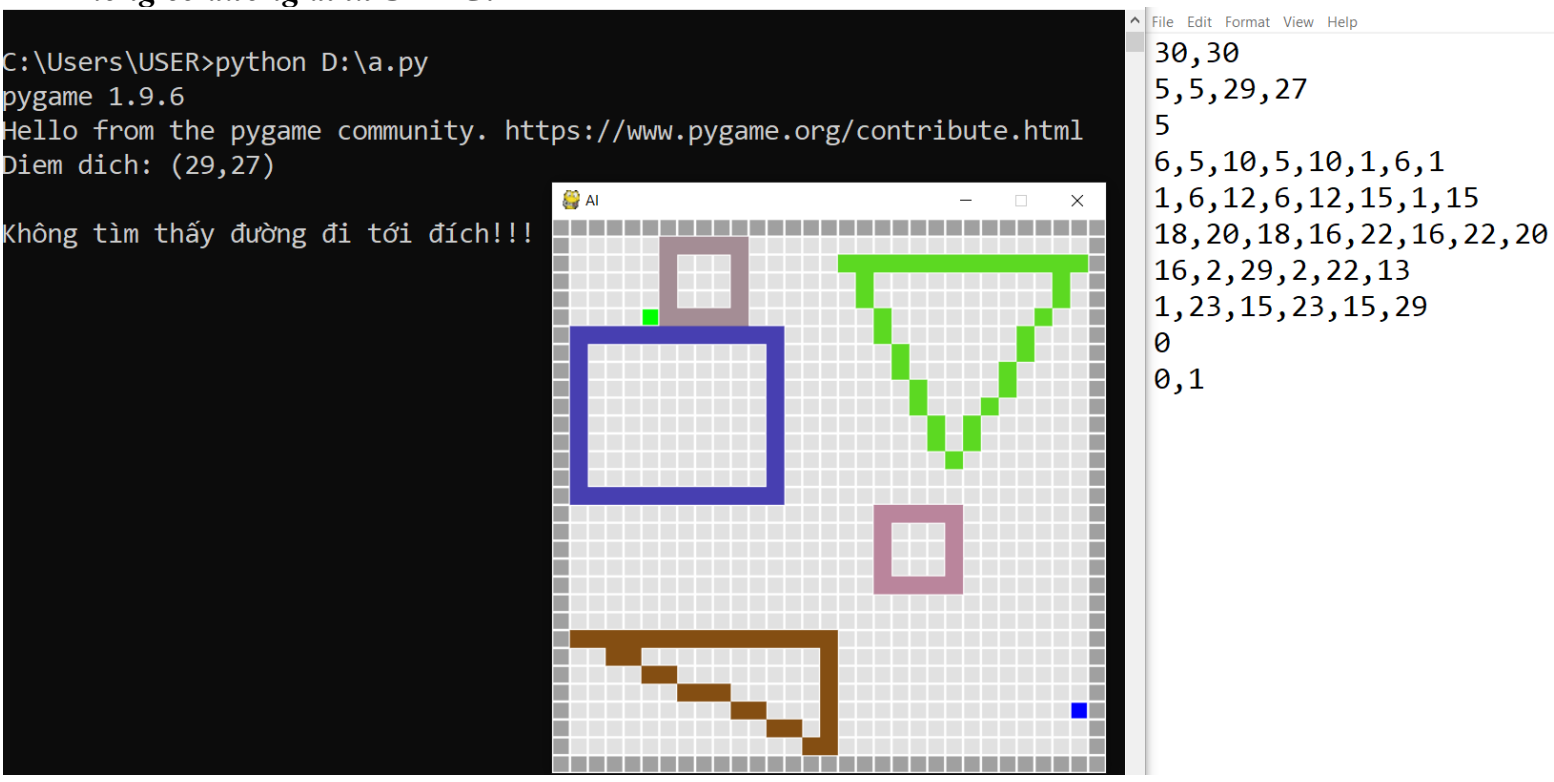
```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (68,69)
(10,10)->(10,11)->(10,12)->(10,13)->(9,13)->(8,13)->(8,14)->(8,15)->(8,16
)->(8,17)->(9,18)->(10,18)->(11,18)->(12,18)->(13,18)->(14,18)->(15,18)->
(16,18)->(17,18)->(18,18)->(19,18)->(20,18)->(21,18)->(22,18)->(23,18)->(
24,18)->(25,18)->(26,18)->(27,18)->(28,18)->(29,19)->(29,20)->(29,21)->(2
9,22)->(29,23)->(29,24)->(29,25)->(29,26)->(29,27)->(29,28)->(29,29)->(29
,30)->(29,31)->(29,32)->(29,33)->(29,34)->(29,35)->(29,36)->(29,37)->(29,
38)->(29,39)->(29,40)->(29,41)->(29,42)->(30,43)->(31,44)->(32,45)->(33,4
6)->(34,47)->(35,48)->(36,48)->(37,48)->(38,48)->(39,48)->(40,48)->(41,48
)->(42,48)->(43,48)->(44,48)->(45,48)->(46,48)->(47,48)->(48,48)->(49,48)
->(50,48)->(51,48)->(52,48)->(53,48)->(54,48)->(55,48)->(56,48)->(57,48)-
>(58,48)->(59,48)->(60,48)->(61,48)->(62,48)->(63,48)->(64,48)->(65,48)->
(66,48)->(67,48)->(68,49)->(69,50)->(69,51)->(69,52)->(69,53)->(69,54)->(
69,55)->(69,56)->(69,57)->(69,58)->(69,59)->(69,60)->(69,61)->(69,62)->(6
9,63)->(69,64)->(69,65)->(69,66)->(69,67)->(69,68)->(69,69)->(68,69)
Độ dài đường đi:  119.0
```

File Edit Format View Help

```
70,70
10,10,68,69
7
1,3,1,6,7,6,7,3
9,14,14,14,14,17,9,17
40,30,40,40,35,47,35,40,35,30
12,19,12,30,20,35,28,30,28,19
20,15,30,15,30,3
45,4,55,4,55,12,45,12
42,68,68,68,68,50,42,50
0
0,1
```



Không có đường đi từ S => G:



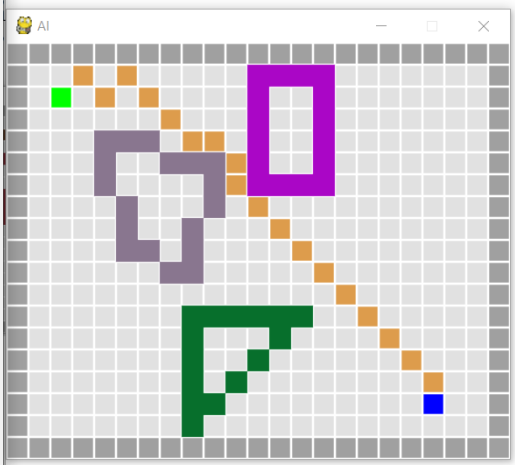
b. <u>BFS:</u>

Có đường đi từ S => G:

<u>Ví dụ 1:</u>

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (19,16)
(2,2)->(2,2)->(3,1)->(4,2)->(5,1)->(6,2)->(7,3)->(8,4)->(9,4)->(10,5)->(
10,6)->(11,7)->(12,8)->(13,9)->(14,10)->(15,11)->(16,12)->(17,13)->(18,1
4)->(19,15)->(19,16)
Độ dài đường đi:  29.0
```

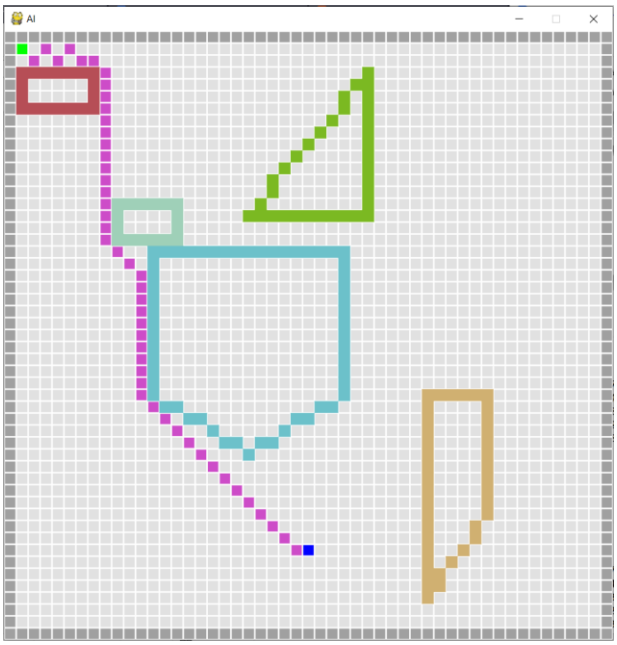
```
File  Edit  Format  View  Help
22,18
2,2,19,16
3
4,4,5,9,8,10,9,5
8,12,8,17,13,12
11,1,11,6,14,6,14,1
0
0,2
```



Ví dụ 2:

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (25,43)
(1,1)->(1,1)->(2,2)->(3,1)->(4,2)->(5,1)->(6,2)->(7,2)->(8,3)->(8,4)->(8,
5)->(8,6)->(8,7)->(8,8)->(8,9)->(8,10)->(8,11)->(8,12)->(8,13)->(8,14)->(
8,15)->(8,16)->(8,17)->(9,18)->(10,19)->(11,20)->(11,21)->(11,22)->(11,23
)->(11,24)->(11,25)->(11,26)->(11,27)->(11,28)->(11,29)->(11,30)->(12,31)
->(13,32)->(14,33)->(15,34)->(16,35)->(17,36)->(18,37)->(19,38)->(20,39)-
>(21,40)->(22,41)->(23,42)->(24,43)->(25,43)
Độ dài đường đi:  61.0
```

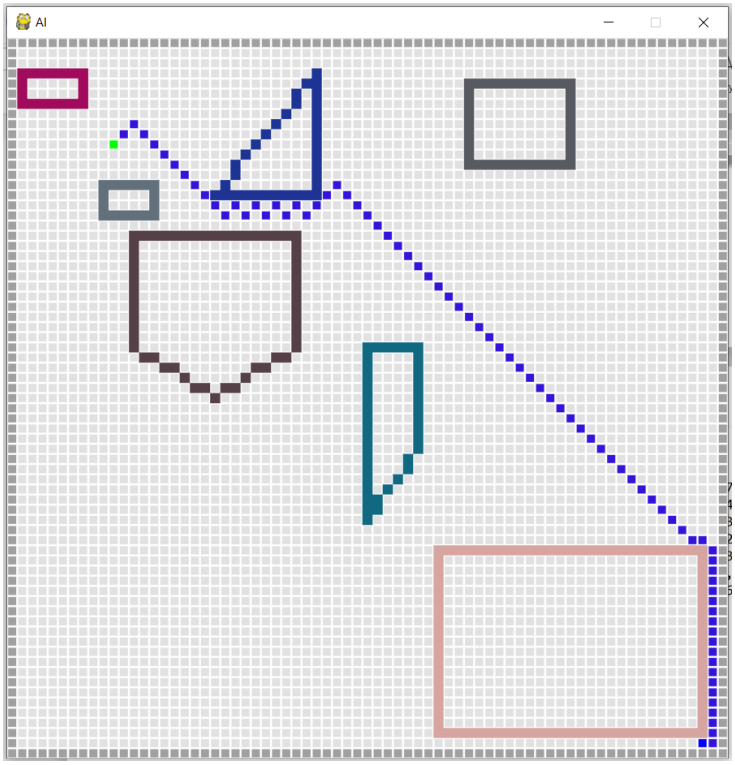
```
File  Edit  Format  View  Help
50,50
1,1,25,43
5
1,3,1,6,7,6,7,3
9,14,14,14,14,17,9,17
40,30,40,40,35,47,35,40,35,30
12,18,12,30,20,35,28,30,28,18
20,15,30,15,30,3
0
0,2
```



Ví dụ 3:

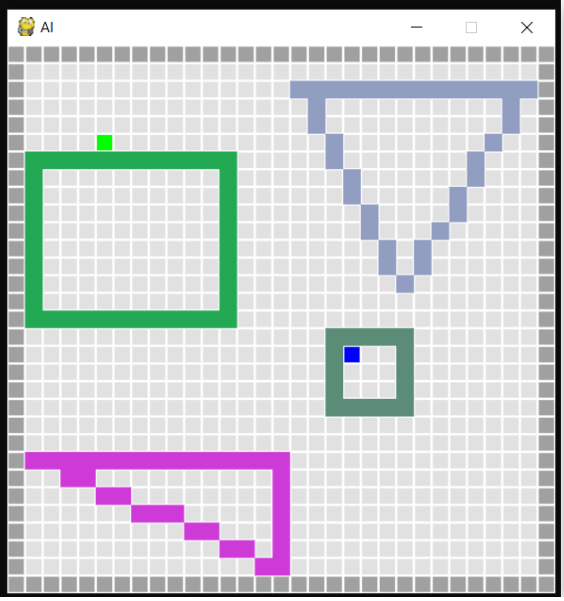
```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (68,69)
(10,10)->(10,10)->(11,9)->(12,8)->(13,9)->(14,10)->(15,11)->(16,12)->(17,
13)->(18,14)->(19,15)->(20,16)->(21,17)->(22,16)->(23,17)->(24,16)->(25,1
7)->(26,16)->(27,17)->(28,16)->(29,17)->(30,16)->(31,15)->(32,14)->(33,15
)->(34,16)->(35,17)->(36,18)->(37,19)->(38,20)->(39,21)->(40,22)->(41,23)
->(42,24)->(43,25)->(44,26)->(45,27)->(46,28)->(47,29)->(48,30)->(49,31)-
>(50,32)->(51,33)->(52,34)->(53,35)->(54,36)->(55,37)->(56,38)->(57,39)->
(58,40)->(59,41)->(60,42)->(61,43)->(62,44)->(63,45)->(64,46)->(65,47)->(
66,48)->(67,49)->(68,49)->(69,50)->(69,51)->(69,52)->(69,53)->(69,54)->(6
9,55)->(69,56)->(69,57)->(69,58)->(69,59)->(69,60)->(69,61)->(69,62)->(69
,63)->(69,64)->(69,65)->(69,66)->(69,67)->(69,68)->(69,69)->(68,69)
Độ dài đường đi:  110.0
```

```
File  Edit  Format  View  Help
70,70
10,10,68,69
7
1,3,1,6,7,6,7,3
9,14,14,14,14,17,9,17
40,30,40,40,35,47,35,40,35,30
12,19,12,30,20,35,28,30,28,19
20,15,30,15,30,3
45,4,55,4,55,12,45,12
42,68,68,68,68,50,42,50
0
0,2
```



Không có đường đi từ S => G:

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (19,17)

Không tìm thấy đường đi tới đích!!!
```




```
30,30
5,5,19,17
4
1,6,12,6,12,15,1,15
18,20,18,16,22,16,22,20
16,2,29,2,22,13
1,23,15,23,15,29
0
0,2
```

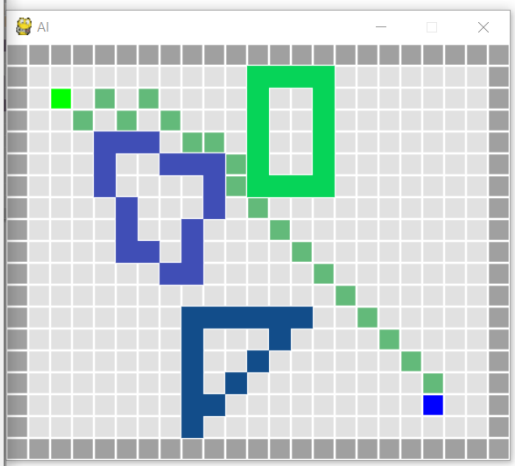
c. *A*:*

Có đường đi từ S => G:

Ví dụ 1:

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (19,16)
(2,2)->(2,2)->(3,3)->(4,2)->(5,3)->(6,2)->(7,3)->(8,4)->(9,4)->(10,5)->(
10,6)->(11,7)->(12,8)->(13,9)->(14,10)->(15,11)->(16,12)->(17,13)->(18,1
4)->(19,15)->(19,16)
Độ dài đường đi:  29.0
```

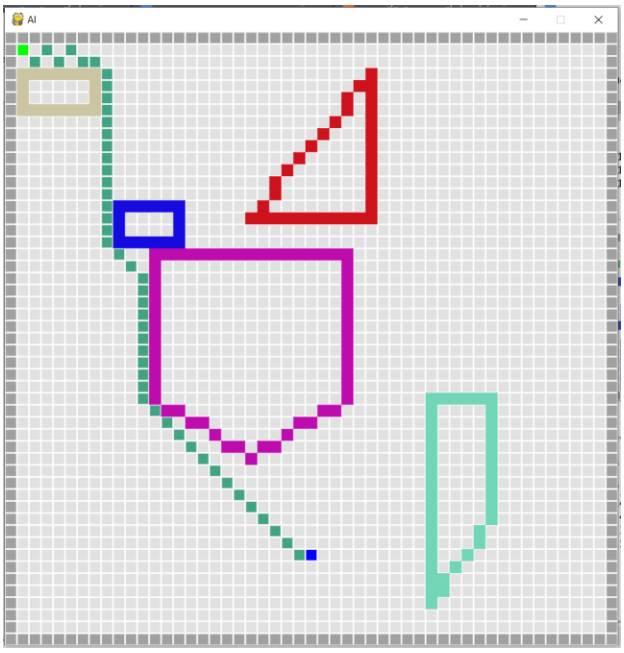
```
File  Edit  Format  View  Help
22,18
2,2,19,16
3
4,4,5,9,8,10,9,5
8,12,8,17,13,12
11,1,11,6,14,6,14,1
0
0,3
```



Ví dụ 2:

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (25,43)
(1,1)->(1,1)->(2,2)->(3,1)->(4,2)->(5,1)->(6,2)->(7,2)->(8,3)->(8,4)->(8,
5)->(8,6)->(8,7)->(8,8)->(8,9)->(8,10)->(8,11)->(8,12)->(8,13)->(8,14)->(
8,15)->(8,16)->(8,17)->(9,18)->(10,19)->(11,20)->(11,21)->(11,22)->(11,23
)->(11,24)->(11,25)->(11,26)->(11,27)->(11,28)->(11,29)->(11,30)->(12,31)
->(13,32)->(14,33)->(15,34)->(16,35)->(17,36)->(18,37)->(19,38)->(20,39)-
>(21,40)->(22,41)->(23,42)->(24,43)->(25,43)
Độ dài đường đi:  61.0
```

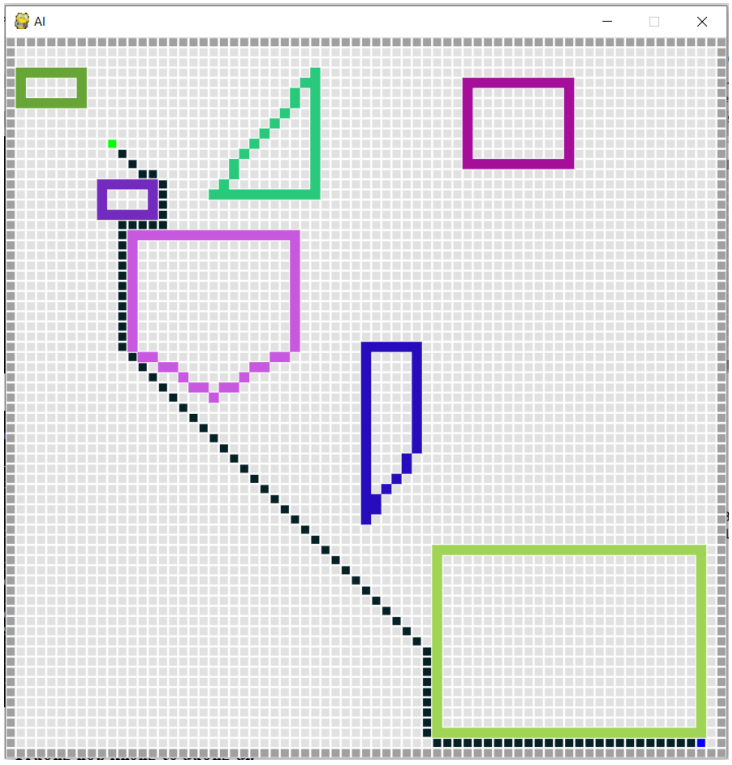
```
File  Edit  Format  View  Help
50,50
1,1,25,43
5
1,3,1,6,7,6,7,3
9,14,14,14,14,17,9,17
40,30,40,40,35,47,35,40,35,30
12,18,12,30,20,35,28,30,28,18
20,15,30,15,30,3
0
0,3
```



Ví dụ 3:

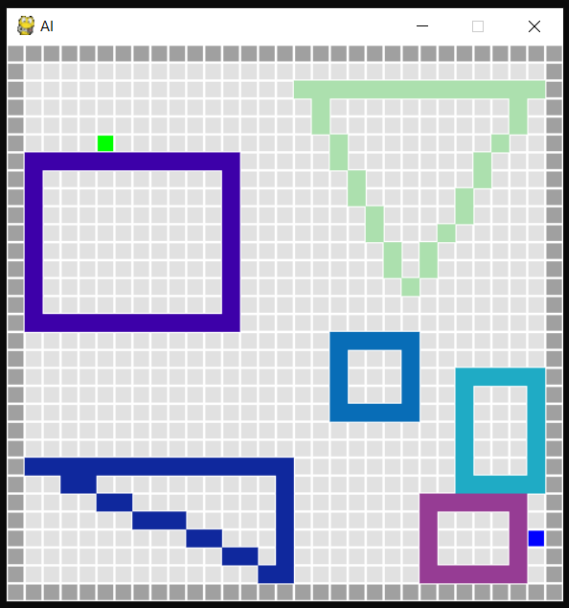
```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (68,69)
(10,10)->(10,10)->(11,11)->(12,12)->(13,13)->(14,13)->(15,14)->(15,15)->(
15,16)->(15,17)->(15,18)->(14,18)->(13,18)->(12,18)->(11,18)->(11,19)->(1
1,20)->(11,21)->(11,22)->(11,23)->(11,24)->(11,25)->(11,26)->(11,27)->(11
,28)->(11,29)->(11,30)->(12,31)->(13,32)->(14,33)->(15,34)->(16,35)->(17,
36)->(18,37)->(19,38)->(20,39)->(21,40)->(22,41)->(23,42)->(24,43)->(25,4
4)->(26,45)->(27,46)->(28,47)->(29,48)->(30,49)->(31,50)->(32,51)->(33,52
)->(34,53)->(35,54)->(36,55)->(37,56)->(38,57)->(39,58)->(40,59)->(41,60)
->(41,61)->(41,62)->(41,63)->(41,64)->(41,65)->(41,66)->(41,67)->(41,68)-
>(42,69)->(43,69)->(44,69)->(45,69)->(46,69)->(47,69)->(48,69)->(49,69)->
(50,69)->(51,69)->(52,69)->(53,69)->(54,69)->(55,69)->(56,69)->(57,69)->(
58,69)->(59,69)->(60,69)->(61,69)->(62,69)->(63,69)->(64,69)->(65,69)->(6
6,69)->(67,69)->(68,69)
Độ dài đường đi:  109.5
```

```
File  Edit  Format  View  Help
70,70
10,10,68,69
7
1,3,1,6,7,6,7,3
9,14,14,14,14,17,9,17
40,30,40,40,35,47,35,40,35,30
12,19,12,30,20,35,28,30,28,19
20,15,30,15,30,3
45,4,55,4,55,12,45,12
42,68,68,68,68,50,42,50
0
0,3
```



Không có đường đi từ S => G:

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Diem dich: (29,27)

Không tìm thấy đường đi tới đích!!!
```




```
File  Edit  Format  View  Help
30,30
5,5,29,27
6
1,6,12,6,12,15,1,15
18,20,18,16,22,16,22,20
16,2,29,2,22,13
1,23,15,23,15,29
28,29,28,25,23,25,23,29
29,24,25,24,25,18,29,18
0
0,3
```

*** Nhận xét sự khác nhau khi chạy 3 thuật toán:**

Nội dung	BFS	Greedy Best-First Search	A*
Loại tìm kiếm	Tìm kiếm mù	Tìm kiếm có thông tin với hàm Heuristic *h(x)* là chi phí để đi từ x đến nút đích	Tìm kiếm có thông tin sử dụng hàm Heuristic *h(x) = f(x) + g(x)* với f(x) là khoảng cách từ x tới GOAL và g(x) là khoảng cách từ START tới x

- Lý do 3 thuật toán cho ra 3 kết quả khác nhau là do quá trình duyệt khác nhau:
  + BFS: Ta bắt đầu từ đỉnh gốc và lần lượt nhìn các đỉnh kề với đỉnh gốc. Sau đó, với mỗi đỉnh trong số đó, thuật toán lại lần lượt nhìn trước các đỉnh kề với nó mà chưa được quan sát trước đó và lặp lại.
  + A* và Greedy thuật toán cũng tương tự, tuy nhiên đỉnh để quan sát tiếp theo không phải đỉnh ở đầu hàng đợi như BFS mà là đỉnh tốt nhất theo tiêu chí của mỗi thuật toán (Dựa trên hàm *heuristic* của mỗi thuật toán).
    ~ Thuật toán Greedy hàm *heuristic* dựa trên tiêu chí "tham lam" đỉnh tiếp theo được chọn sẽ làm điểm gần đích nhất. Thuật toán này không quan tâm đến chi phí của đường đi tổng thể mà chỉ quan tâm đến đỉnh đến tiếp theo.
    ~ Thuật toán A* ngoài tiêu chí như Greedy nó còn tính đến khoảng cách đã đi qua, điều đó làm cho A* "đầy đủ" và "tối ưu".
- So sánh tốc độ thực thi thì Greedy và A* trung bình sẽ nhanh hơn BFS vì nó không "vét cạn" như BFS mà A* và Greedy chỉ mở rộng trên một điểm nếu điểm đó "tốt" theo tiêu chí *heuristic* của nó . Vì vậy nó chỉ tập trung vào việc tiếp cận điểm mục tiêu càng nhanh càng tốt từ nút hiện tại, chứ không phải cố gắng tiếp cận mọi điểm khác.

3. Mức 3: Trên bản đồ sẽ xuất hiện thêm một số điểm khác được gọi là điểm đón

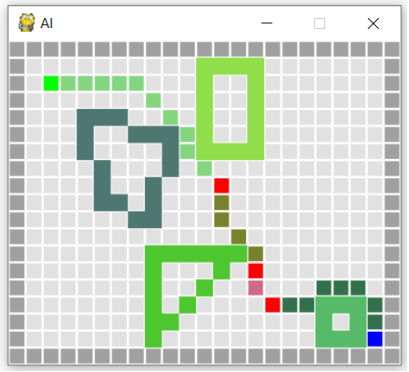
- Thực hiện tìm kiếm tham lam. Từ điểm bắt đầu, tìm điểm đón nào gần mình nhất (nếu nhiều điểm có cùng khoảng cách ngắn nhất ta xét đến khoảng cách từ điểm đó đến đích để ưu tiên duyệt) và gán điểm đó là GOAL, sau đó thực hiện việc tìm kiếm đến điểm đón đã chọn.
- Khi đã duyệt qua hết tất cả các điểm đón, ta tìm đường đi đến điểm đích. Nếu không thể tìm đường đi đến điểm đón thì thông báo ra màn hình.
- Chi phí đường đi được tính như Level 1

Có đường đi từ S => G:

Ví dụ 1:

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Điểm cần đón:  (12,8)
(2,2)->(3,2)->(4,2)->(5,2)->(6,2)->(7,2)->(8,3)->(9,4)->(10,5)->(10,6)->(
11,7)->(12,8)

Điểm cần đón:  (14,13)
(12,8)->(12,9)->(12,10)->(13,11)->(14,12)->(14,13)

Điểm cần đón:  (15,15)
(14,13)->(14,14)->(15,15)

Diem dich: (21,17)
(15,15)->(16,15)->(17,15)->(18,14)->(19,14)->(20,14)->(21,15)->(21,16)->(
21,17)
Độ dài đường đi:  32.0
```

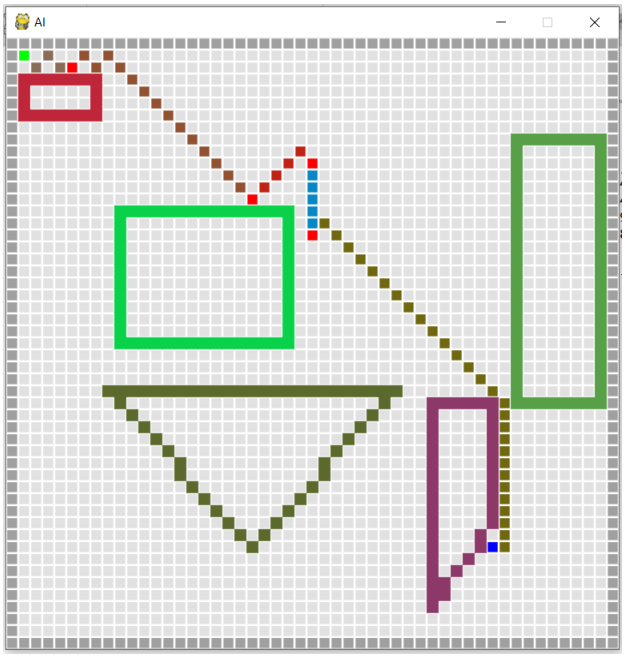
```
File Edit Format View Help
22,18
2,2,21,17
4
4,4,5,9,8,10,9,5
8,12,8,17,13,12
11,1,11,6,14,6,14,1
18,17,18,15,20,15,20,17
3
15,15,14,13,12,8
0,1
```



Ví dụ 2:

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Điểm cần đón:  (5,2)
(1,1)->(1,1)->(2,2)->(3,1)->(4,2)->(5,2)

Điểm cần đón:  (20,13)
(5,2)->(1,1)->(6,1)->(7,2)->(8,1)->(9,2)->(10,3)->(11,4)->(12,5)->(13,6)-
>(14,7)->(15,8)->(16,9)->(17,10)->(18,11)->(19,12)->(20,13)

Điểm cần đón:  (25,10)
(20,13)->(1,1)->(21,12)->(22,11)->(23,10)->(24,9)->(25,10)

Điểm cần đón:  (25,16)
(25,10)->(1,1)->(25,11)->(25,12)->(25,13)->(25,14)->(25,15)->(25,16)

Diem dich: (40,42)
(25,16)->(1,1)->(26,15)->(27,16)->(28,17)->(29,18)->(30,19)->(31,20)->(32
,21)->(33,22)->(34,23)->(35,24)->(36,25)->(37,26)->(38,27)->(39,28)->(40,
29)->(41,30)->(41,31)->(41,32)->(41,33)->(41,34)->(41,35)->(41,36)->(41,3
7)->(41,38)->(41,39)->(41,40)->(41,41)->(41,42)->(40,42)
Độ dài đường đi:  83.0
```

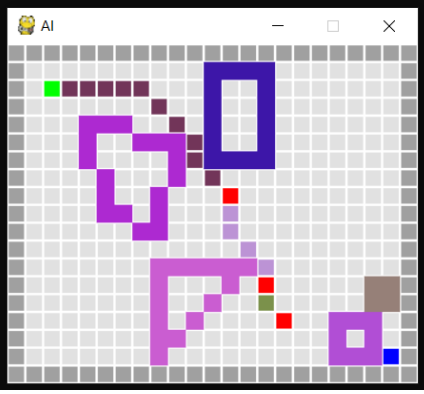
```
File Edit Format View Help
50,50
1,1,40,42
5
1,3,1,6,7,6,7,3
9,14,23,14,23,25,9,25
40,30,40,40,35,47,35,40,35,30
8,29,32,29,20,42
42,30,42,8,49,8,49,30
4
5,2,20,13,25,10,25,16
0,2
```



Không có đường đi từ S => G:

Ví dụ 1:

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Điểm cần đón:  (12,8)
(2,2)->(3,2)->(4,2)->(5,2)->(6,2)->(7,2)->(8,3)->(9,4)->(10,5)->(10,6)->(
11,7)->(12,8)

Điểm cần đón:  (14,13)
(12,8)->(12,9)->(12,10)->(13,11)->(14,12)->(14,13)

Điểm cần đón:  (15,15)
(14,13)->(14,14)->(15,15)

Diem dich: (21,17)

Không tìm thấy đường đi tới đích!!!
```

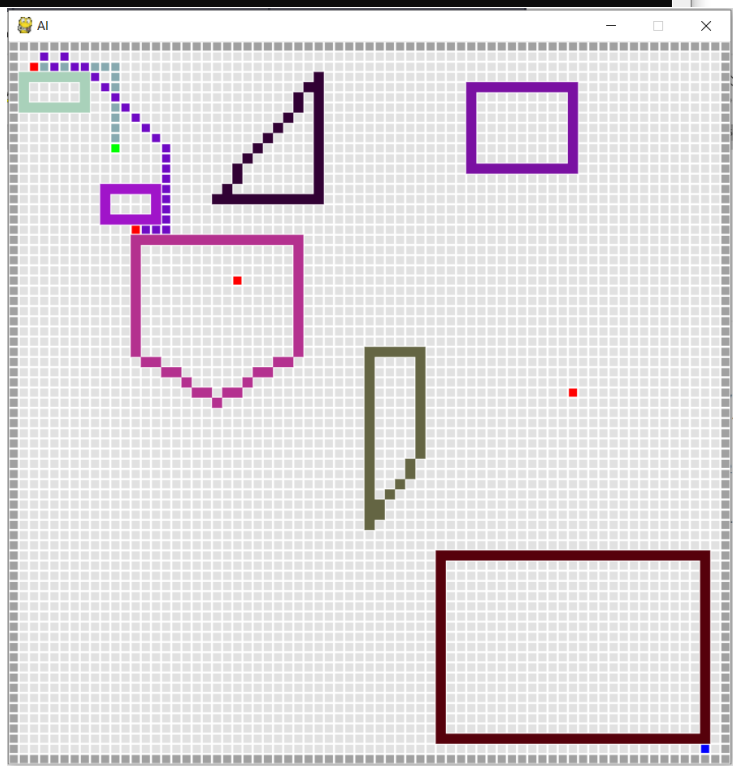



```
File  Edit  Format  View  Help
22,18
2,2,21,17
5
4,4,5,9,8,10,9,5
8,12,8,17,13,12
11,1,11,6,14,6,14,1
18,17,18,15,20,15,20,17
20,13,21,13,21,14,20,14
3
15,15,14,13,12,8
0,1
```

Ví dụ 2:

```
C:\Users\USER>python D:\a.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
Điểm cần đón:  (2,2)
(10,10)->(10,10)->(10,9)->(10,8)->(10,7)->(10,6)->(10,5)->(10,4)->(10,3)->(10,2)->(9,2)->(8,2)->(7,2)->(6,2)->(5,2)->(4,2)->(3,2)->(2,2)

Điểm cần đón:  (12,18)
(2,2)->(10,10)->(3,1)->(4,2)->(5,1)->(6,2)->(7,2)->(8,3)->(9,4)->(10,5)->(11,6)->(12,7)->(13,8)->(14,9)->(15,10)->(15,11)->(15,12)->(15,13)->(15,14)->(15,15)->(15,16)->(15,17)->(15,18)->(14,18)->(13,18)->(12,18)

Điểm cần đón:  (22,23)

Không tìm thấy đường đi tới điểm đón!

Không tìm thấy đường đi tới đích!!!
```

```
File  Edit  Format  View  Help
70,70
10,10,68,69
7
1,3,1,6,7,6,7,3
9,14,14,14,14,17,9,17
40,30,40,40,35,47,35,40,35,30
12,19,12,30,20,35,28,30,28,19
20,15,30,15,30,3
45,4,55,4,55,12,45,12
42,68,68,68,68,50,42,50
4
2,2,22,23,12,18,55,34
0,2
```



4. Mức 4: Các hình đa giác có thể di động

- Khi cho người di chuyển 1 đơn vị pixel thì các đa giác có thể đứng yên hoặc di chuyển ngẫu nhiên ngang/dọc/xéo 1 pixel. Ở Level này ta không xét đến việc đi qua các điểm đón
- Ta cho các đa giác di chuyển trước (cho các đa giác di chuyển ngẫu nhiên và kiểm tra đụng độ, tiếp xúc trước khi di chuyển), sau đó thực hiện thuật toán tìm đường đi cho người đến đích
- Khi không tìm thấy đường đi đến đích sẽ thông báo ra màn hình
- Chi phí đường đi được tính như Level 1

Link video: https://youtu.be/EXvxEmZNMTM

VI. Hướng dẫn chạy:

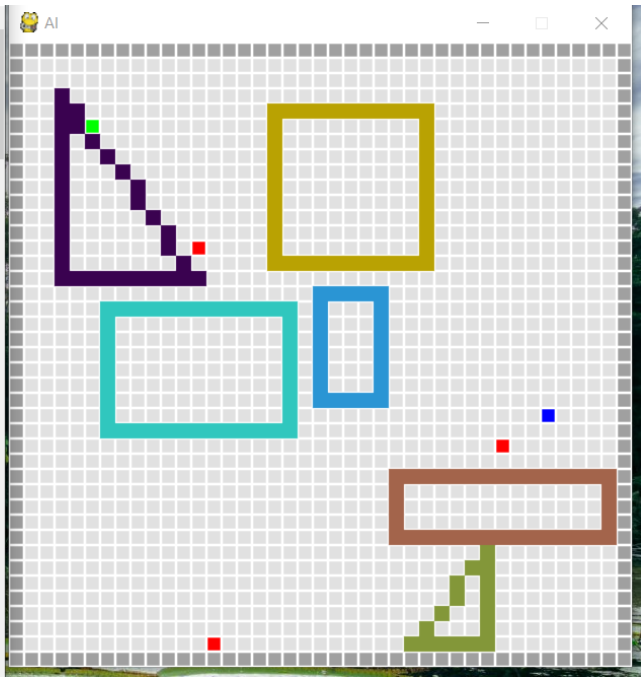
1. Cấu trúc file Input:

```
CONST_SCALE = 12

# Mở file
f = open("D:\Input.txt")
```

Trước khi chạy thuật toán, ta có thể điều chỉnh kích thước của 1 pixel thông qua hằng số *CONST_SCALE*
Thay đổi đường dẫn đến file Input ở hàm *f = open("Link_of_Input_File")*
- Dòng đầu tiên: Kích thước giới hạn của không gian với chiều ngang và chiều cao
- Dòng thứ hai: Lần lượt là điểm bắt đầu và điểm kết thúc
- Dòng thứ ba: N - số đa giác trong không gian
- N dòng tiếp theo: Mỗi dòng lần lượt là danh sách các đỉnh của đa giác theo chiều kim đồng hồ
- Dòng tiếp theo: M - số điểm đón
- Dòng tiếp theo: Danh sách các điểm đón
- Dòng cuối: Gồm 2 số:
    + Số thứ nhất: 0 hoặc 1 - tương ứng với đa giác đứng yên hay di chuyển
    + Số thứ hai: 1, 2, 3 - tương ứng với thuật toán Greedy, BFS hay A*

Ví dụ: File bên dưới bên dưới có:
- Kích thước chiều ngang là 40 và chiều dọc là 40
- Người sẽ bắt đầu từ điểm (5,5) và đi đến đích là điểm (35,24)
- Có 6 đa giác như hình
- Có 3 điểm đón là (12,13), (13,39), (32,26)
- Và ví dụ này đang xét trong không gian mà các đa giác đứng yên và thuật toán tìm đường đi được thực hiện là thuật toán Greedy

```
40,40
5,5,35,24
6
6,17,6,25,18,25,18,17
17,4,17,14,27,14,27,4
20,16,24,16,24,23,20,23
3,3,3,15,12,15
25,32,39,32,39,28,25,28
31,39,26,39,31,33
3
12,13,13,39,32,26
0,1
```



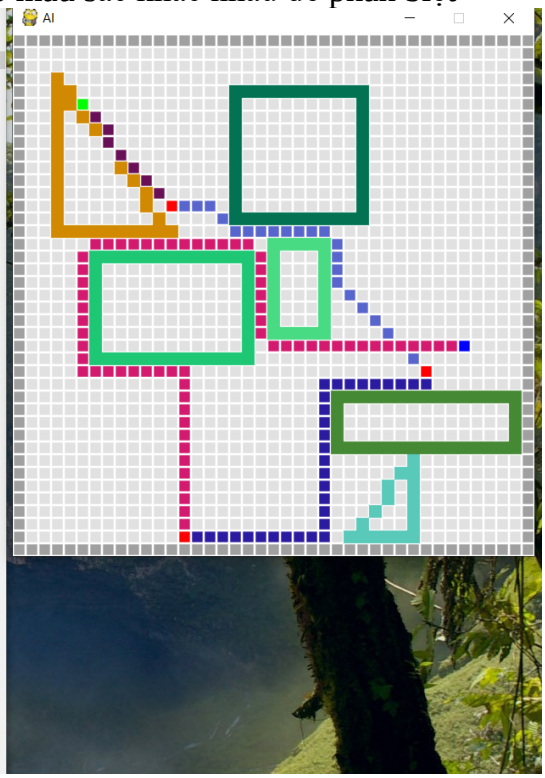
2. Cấu trúc ouput:

- Nếu có đường đi từ S => G và đi qua được tất cả các điểm đón thì lộ trình đường đi sẽ được in ra màn hình Console cùng với chi phí đường đi đó (tổng các pixel đã đi qua với +1 cho bước đi ngang hay dọc và +1.5 cho bước đi chéo)
- Nếu không có đường đi từ S => hoặc không thể đến được các điểm đón thì sẽ thông báo ra màn hình Console: "Không tìm thấy đường đi đến đích" hay "Không tìm thấy đường đi tới điểm đón"

Ví dụ:

Lộ trình đường đi từ S => G qua các điểm đón có tổng chi phí là 140 với
- Điểm bắt đầu (5,5) có màu xanh lá
- Điểm kết thúc (35,24) có màu xanh dương
- Các điểm đón có màu đỏ
- Thứ tự đến thăm các điểm đón được in ra màn hình lần lượt là điểm (5,5) => (12,13) => (32,36) => (13,39) => (35,24)
- Mỗi tuyến đường đi từ điểm hiện tại tới điểm đón hay tới đích sẽ có màu sắc khác nhau để phân biệt

```
Điểm cần đón:  (12,13)
(5,5)->(6,6)->(7,7)->(7,8)->(8,9)->(9,10)->(10,11)->(11,12)->(12,13)

Điểm cần đón:  (32,26)
(12,13)->(13,13)->(14,13)->(15,13)->(16,14)->(17,15)->(18,15)->(19,15)->(
20,15)->(21,15)->(22,15)->(23,15)->(24,15)->(25,16)->(25,17)->(25,18)->(2
5,19)->(26,20)->(27,21)->(28,22)->(29,23)->(30,24)->(31,25)->(32,26)

Điểm cần đón:  (13,39)
(32,26)->(32,27)->(31,27)->(30,27)->(29,27)->(28,27)->(27,27)->(26,27)->(
25,27)->(24,27)->(24,28)->(24,29)->(24,30)->(24,31)->(24,32)->(24,33)->(2
4,34)->(24,35)->(24,36)->(24,37)->(24,38)->(24,39)->(23,39)->(22,39)->(21
,39)->(20,39)->(19,39)->(18,39)->(17,39)->(16,39)->(15,39)->(14,39)->(13,
39)

Diem dich: (35,24)
(13,39)->(13,38)->(13,37)->(13,36)->(13,35)->(13,34)->(13,33)->(13,32)->(
13,31)->(13,30)->(13,29)->(13,28)->(13,27)->(13,26)->(12,26)->(11,26)->(1
0,26)->(9,26)->(8,26)->(7,26)->(6,26)->(5,26)->(5,25)->(5,24)->(5,23)->(5
,22)->(5,21)->(5,20)->(5,19)->(5,18)->(5,17)->(6,16)->(7,16)->(8,16)->(9,
16)->(10,16)->(11,16)->(12,16)->(13,16)->(14,16)->(15,16)->(16,16)->(17,1
6)->(18,16)->(19,17)->(19,18)->(19,19)->(19,20)->(19,21)->(19,22)->(19,23
)->(20,24)->(21,24)->(22,24)->(23,24)->(24,24)->(25,24)->(26,24)->(27,24)
->(28,24)->(29,24)->(30,24)->(31,24)->(32,24)->(33,24)->(34,24)->(35,24)
Độ dài đường đi:  140.0
```



VII. Mã nguồn tham khảo:

https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/
https://vi.wikipedia.org/wiki/T%C3%ACm_ki%E1%BA%BFm_theo_chi%E1%BB%81u_r%E1%BB%99ng
http://www.ariel.com.au/a/python-point-int-poly.html
https://www.pygame.org/docs/ref/draw.html
https://sites.cs.ucsb.edu/~pconrad/cs5nm/topics/pygame/drawing/
https://www.youtube.com/watch?v=g4E9iq0BixA