



KHOA CÔNG NGHỆ THÔNG TIN
ĐỒ ÁN MÔN HỌC
CƠ SỞ TRÍ TUỆ NHÂN TẠO

THUẬT TOÁN TÌM KIẾM

Thành viên:

Nguyễn Minh Lương – 19120571

Huỳnh Nguyễn Thị Lựu – 19120573

So sánh các thuật toán trên các bản đồ không có điểm thưởng:

Bản đồ 1:

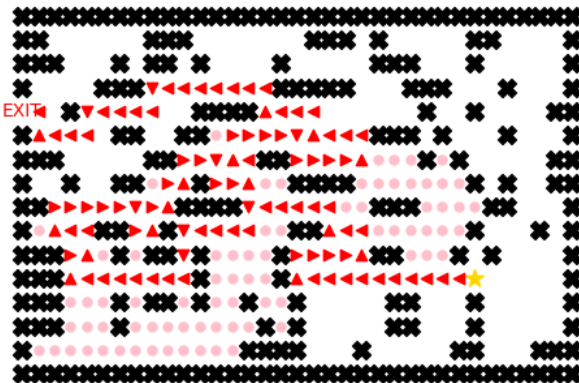


Không có điểm thưởng.

Điểm xuất phát đặt tại (11, 28).

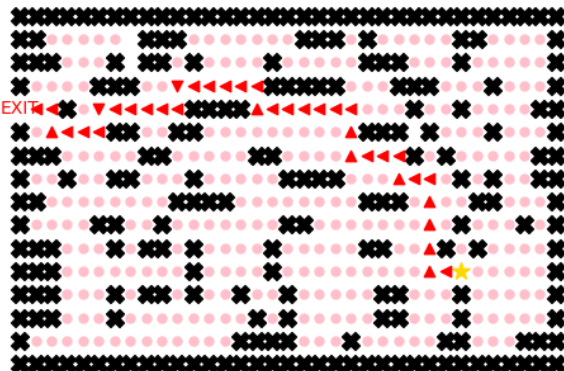
Và điểm thoát đặt tại (4, 0).

Thuật toán Depth First Search (DFS):



Với thuật toán DFS, đường đi tìm được có chi phí là **101**.

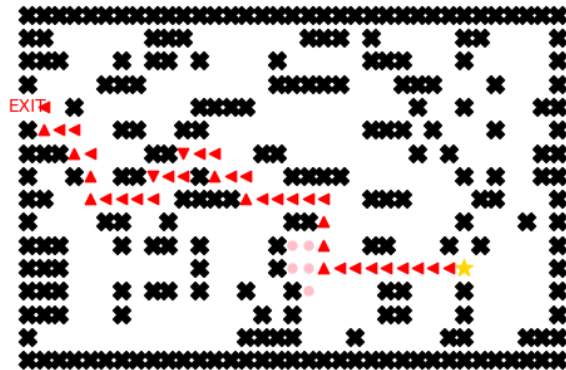
Thuật toán Breadth First Search (BFS):



Đường đi tìm được có chi phí là **39**.

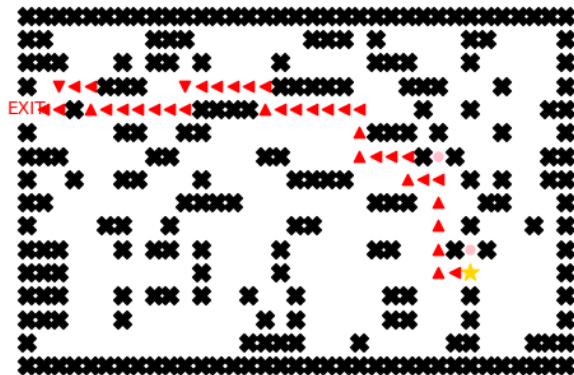
Thuật toán Greedy Best First Search (GBFS):

Sử dụng Euclid Norm:



Đường đi tìm được có chi phí là 39.

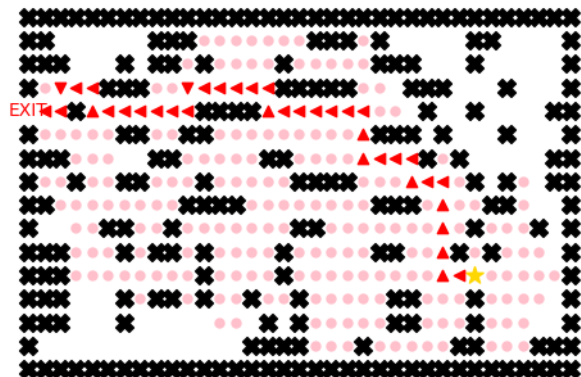
Sử dụng Manhattan Norm:



Đường đi tìm được có chi phí là 39.

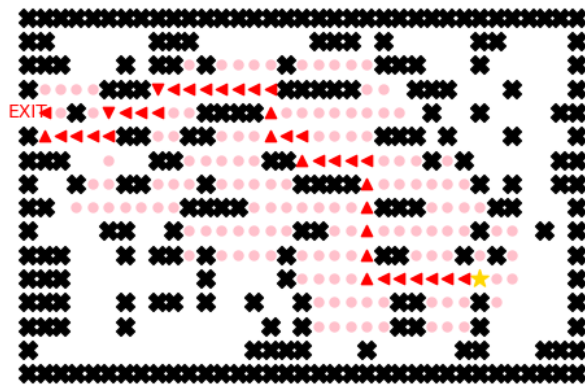
Thuật toán A*:

Sử dụng Euclid Norm:



Đường đi tìm được có chi phí là 39.

Sử dụng Manhattan Norm:



Đường đi tìm được có chi phí là 39.

Nhận xét:

- Đối với bản đồ thứ 1, thuật toán GBFS sử dụng Manhattan là thuật toán tìm được đường đi tốt nhất, với chi phí đường đi là 39 và các điểm được duyệt ít nhất.
- A* và GBFS đều cho ra đường đi tối ưu.
- Thuật toán A* là thuật toán vừa hoàn thiện vừa tối ưu nên sẽ đảm bảo tìm được đường đi tối ưu (nếu có trong trường hợp hàm Heuristic phải hợp lý, nhất quán).
- Còn GBFS thì tùy thuộc vào từng Loại Heuristic và từng Loại bản đồ thì có thể tìm ra được lời giải tốt nhất.
- Nhưng riêng DFS thì không, DFS tìm được đường đi có chi phí rất lớn là 101 với số Lượng Lớn điểm được duyệt. Vì DFS là thuật toán không tối ưu nó chỉ tìm được đường đi chứ không tìm được đường đi ngắn nhất (tùy trường hợp).
- Thuật toán BFS cũng cho ra được đường đi tối ưu nhưng số điểm đã duyệt qua thì rất Lớn (toàn bản đồ). Thuật toán BFS tối ưu khi chi phí của mỗi điểm là như nhau.
- với 2 thuật toán tìm kiếm có thông tin là GBFS có phần tốt hơn vì vừa tìm được đường đi tối ưu vừa có số điểm duyệt qua ít nhất, trong cả 2 thuật toán này thì khi sử dụng Hàm Heuristic Là Manhattan Norm đều cho kết quả tốt hơn hàm Heuristic còn Lại.

Bản đồ 2:

```

XXXXXXXXXXXXXXXXXXXXXXXXX
X      X      XX      X
X      X      XXXXXXXXXX
X  X      XXXX  XXXXX  XX
EXIT X      X  X  XX      XXXXX  X
X      XXX  XXXX  X  X
XXXXXXXXXX X      XX  X  X
XXXXXXXXXXXXX  X  X  XX      X
X      X      X  X  ★  X  X
XXXXXX  X  X  X  X      X  X
XXXXXXXXXXXXXXXXXXXXXXXXX

```

Không có điểm thưởng.

Điểm xuất phát đặt tại (8, 15).

Và điểm thoát đặt tại (4, 0).

Thuật toán Depth First Search (DFS):

```

XXXXXXXXXXXXXXXXXXXXXXXXX
X      X      XX      X
X      X      XXXXXXXXXX
X  X      XXXX  XXXXX  XX
EXIT ◀ X      X  X  XX      XXXXX  X
X ▲ ◀ ◀ ◀ ◀ ◀ ◀ ◀ XX  XXXX  X  X
XXXXXXXXXX ▲ ◀ ◀ ◀ ◀ ◀ XX  X  X
XXXXXXXXXXXXX ● ● X  ▲ ● XX      X
X ● ● ● ● ● ● ● X  ▲ ★ ● ● X  X
XXXXXXXXX ● ● ● ● ● X  ● ● ● ● ● X
XXXXXXXXXXXXXXXXXXXXXXXXX

```

Đường đi tìm được có chi phí là 19.

Thuật toán Breadth First Search (BFS):

```

XXXXXXXXXXXXXXXXXXXXXXXXX
X      X ● ● ● XX ● XX      X
X      ● ● ● X ● ● ● ● XXXXXXXXXX
X  X ● ● ● ● XX ● ● XXXXX  XXXX
EXIT ◀ X ● ● ● X ● ● XX ● ● XXXXX  X
X ▲ ◀ ◀ ◀ ◀ ◀ ◀ ◀ XX ● XXXX  X  X
XXXXXXXXXX ● X  ▲ ◀ ◀ ◀ ◀ XX      X
XXXXXXXXXXXXX ● ● X  ▲ ● XX      X
X ● ● ● ● ● ● ● X  X  ★ ● ● X  X
XXXXXXXXX ● ● ● ● X  ● ● ● ● ● X
XXXXXXXXXXXXXXXXXXXXXXXXX

```

Đường đi tìm được có chi phí là 19.

Thuật toán Greedy Best First Search (GBFS):

Sử dụng Euclid Norm:

```

*****
*      *      **  **          *
*      *      ****
*  *      ***  ****  **  **
EXIT< *      *  *  *      ****
*  ▲<<<<<<<<<<  **  **  *  *
*****  *  ▲<<<<<<<  **  *  *
*****  *      *  ▲  **  *  *
*      *      *  ▲  *  *  *  *
*****  *      *  *  *      *  *
*****

```

Đường đi tìm được có chi phí là 19.

Sử dụng Manhattan Norm:

```

*****
*      *      **  **          *
*      *      ****
*  *  .  ***  ****  **  **
EXIT< *  ▼<<<<<<<  *  .  *  .  ****
*  ▲<<<<<<<<<<  *  .  ****  *  *
*****  *  ▲<<<<<<<  **  *  *
*****  *      *  ▲<  **  *  *
*      *      *  *  *  *  *  *
*****  *      *  *  *      *  *
*****

```

Đường đi tìm được có chi phí là 21.

Thuật toán A*:

Sử dụng Euclid Norm:

```

*****
*      *      **  **          *
*      *      ****
*  *      ***  ****  **  **
EXIT< *  .  .  .  .  *  .  *  .  ****
*  ▲<<<<<<<<<<  *  .  ****  *  *
*****  *  .  ▲<<<<<<<  **  *  *
*****  *  .  .  .  *  ▲<  **  *  *
*      *      *  *  *  *  *  *
*****  *      *  *  *  .  .  *  *
*****

```

Đường đi tìm được có chi phí là 19.

Sử dụng Manhattan Norm:

```

*****
*      *      *      *      *
*      *      *      *      *
*      *      *      *      *
*      *      *      *      *
EXIT *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *
*      *      *      *      *
*      *      *      *      *
*  *  *  *  *  *  *  *  *  *
*****

```

Đường đi tìm được có chi phí là 19.

Nhận xét:

- Đối với bản đồ thứ 2, thuật toán GBFS sử dụng Euclid Norm là thuật toán tìm được đường đi tốt nhất, với chi phí đường đi là 19 và không duyệt dư 1 điểm nào.
- A* cũng cho ra đường đi tối ưu. Vì thuật toán này là thuật toán vừa hoàn thiện vừa tối ưu nên sẽ đảm bảo tìm được đường đi tối ưu (nếu có trong trường hợp này).
- Cả 2 thuật toán tìm kiếm mù là DFS và BFS đều cho ra đường đi tối ưu với chi phí là 19. Nhưng DFS có phần nhanh hơn BFS vì có số điểm đã duyệt qua ít hơn so với BFS.
- Nhưng đối với thuật toán GBFS sử dụng Manhattan Norm thì có vẻ không được tối ưu khi tìm ra được đường đi có chi phí là 21. Lớn hơn hẳn so với các đường đi mà các thuật toán khác tìm được.
- Ở bản đồ thứ 2 thì đối với 2 thuật toán tìm kiếm có thông tin thì hàm Euclid Norm có vẻ hoạt động tốt hơn hàm Manhattan Norm.

Bản đồ 3:

```

*****
**      ***
*      ***      *
*      ***      *
EXIT *      *      *
*      *      *      *
*      *      *      *
*      *      *      *
*      *      *      *
*      *      *      *
*      *      *      *
*      *      *      *
*      *      *      *
*      *      *      *
*****

```

Không có điểm thưởng.

Điểm xuất phát đặt tại (8, 26).

Và điểm thoát đặt tại (3, 0)

Thuật toán Depth First Search (DFS):

A 2D maze environment for a reinforcement learning task. The maze is bounded by black 'x' characters. The start is marked by a red triangle and the goal by a yellow star. Red arrows indicate the path from start to goal. The word 'EXIT' is written in red at the top left.

Đường đi tìm được có chi phí là **31**.

Thuật toán Breadth First Search (BFS):

Đường đi tìm được có chi phí là **31**.

Thuật toán Greedy Best First Search (GBFS):

Sử dụng Euclid Norm:

Đường đi tìm được có chi phí là **31**.

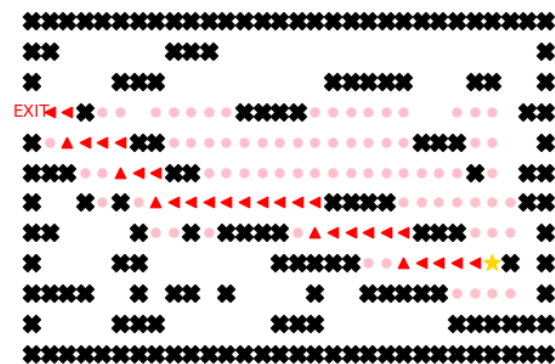
Sử dụng Manhattan Norm:



Đường đi tìm được có chi phí là 35.

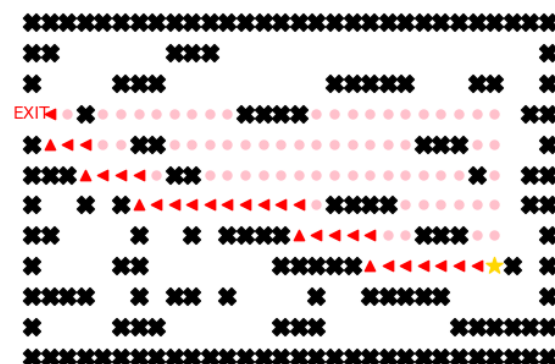
Thuật toán A*:

Sử dụng Euclid Norm:



Đường đi tìm được có chi phí là 31.

Sử dụng Manhattan Norm:



Đường đi tìm được có chi phí là 31.

Nhận xét:

- Đối với bản đồ thứ 3, thuật toán GBFS sử dụng Euclid Norm là thuật toán tìm được đường đi tốt nhất, với chi phí đường đi là 31 và không duyệt dư 1 điểm nào.

- A* cũng cho ra đường đi tối ưu. Vì thuật toán này là thuật toán vừa hoàn thiện vừa tối ưu nên sẽ đảm bảo tìm được đường đi tối ưu (nếu có trong trường hợp này).

- ### Bản đồ 4:

A 15x20 grid world environment. The top row is a red line labeled "EXIT" on the right. The bottom row is a yellow line labeled "START" on the left. The grid contains black obstacles forming a complex maze. The goal is to find a path from the start to the exit.

Điểm đích đặt tại $(0, 19)$.

Thuật toán Depth First Search (DFS):

The grid world is 15 rows by 20 columns. The top row is labeled 'EXIT' at the right end. The bottom row is a solid black line. The grid contains black 'X' marks representing obstacles. Red dots represent the agent's possible positions. Red arrows indicate the direction of movement. A yellow star is located at row 14, column 10.

Đường đi tìm được có chi phí là **71**.

Thuật toán Breadth First Search (BFS):

```
XXXXXXXXXXXXXXXXXXXXXXXXEXITX
X.....>>>>>>>>>>>>>>>X
X.....X.....X.....X.....X
X..X.....XXXXX..X.....X.....X
X..X.....X..X..X.....X.....X
X.....X.....X.....X.....X.....X
XXXX.....X.....X.....X.....X
XXXXX.....X.....X.....X.....X
X.....>>>>>>>>>>>>>>>X
XXXXXXXXX*XX.....X.....X.....X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Đường đi tìm được có chi phí 23.

Thuật toán Greedy Best First Search (GBFS):

Sử dụng Euclid Norm.

```
XXXXXXXXXXXXXXXXXXXXXXXXEXITX
X.....>>>>>>>>>>>>>>>X
X.....X.....X.....X.....X
X..X.....XXXXX..X.....X.....X
X..X.....X..X..X.....X.....X
X.....X.....X.....X.....X.....X
XXXX.....X.....X.....X.....X
XXXXX.....X.....X.....X.....X
X.....>>>>>>>>>>>>>>>X
XXXXXXXXX*XX.....X.....X.....X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Đường đi tìm được có chi phí là 23.

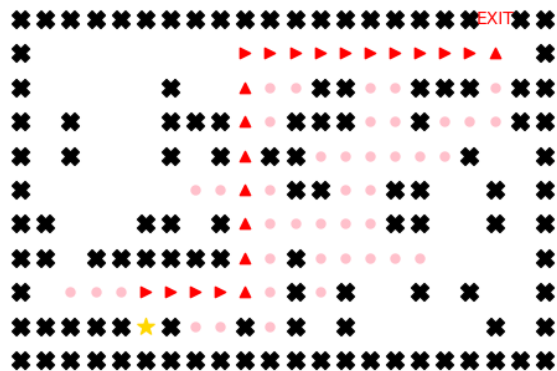
Sử dụng Manhattan Norm.

```
XXXXXXXXXXXXXXXXXXXXXXXXEXITX
X.....>>>>>>>>>>>>>>>X
X.....X.....X.....X.....X
X..X.....XXXXX..X.....X.....X
X..X.....X..X..X.....X.....X
X.....X.....X.....X.....X.....X
XXXX.....X.....X.....X.....X
XXXXX.....X.....X.....X.....X
X.....>>>>>>>>>>>>>>>X
XXXXXXXXX*XX.....X.....X.....X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Đường đi tìm được có chi phí là 23.

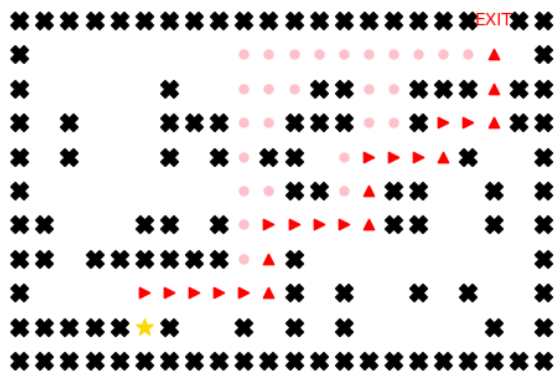
Thuật toán A*:

Sử dụng *Euclid Norm*.



Đường đi tìm được có chi phí là 23.

Sử dụng *Manhattan Norm*.



Đường đi tìm được có chi phí là 23.

Nhận xét:

- Đối với bản đồ thứ 4, thuật toán GBFS sử dụng cả Euclid Norm và Manhattan đều là thuật toán tìm được đường đi tốt nhất, với chi phí đường đi là **23 và không duyệt dư 1 điểm nào**.
- A* cũng cho ra đường đi tối ưu. Vì thuật toán này là thuật toán vừa hoàn thiện vừa tối ưu nên sẽ **đảm bảo tìm được đường đi tối ưu** (nếu có).
- Thuật toán tìm kiếm mù là BFS cũng cho ra đường đi tối ưu với chi phí là **23** vì BFS là thuật toán tìm ra đường đi tối ưu nếu chi phí ở mỗi điểm là như nhau. Tuy nhiên, số lượng điểm mà BFS duyệt qua rất lớn (trong trường hợp này đã duyệt qua gần như hết bản đồ).
- Đối với thuật toán DFS đường đi tìm được có chi phí **71** lớn gấp hơn 3 lần so với đường đi tối ưu. Điều này cũng không có gì đáng ngạc nhiên vì DFS được đánh giá là thuật toán không tối ưu.
- Ở bản đồ thứ 4 thì đối với 2 thuật toán tìm kiếm có thông tin thì **hàm Euclid Norm có vẻ hoạt động tốt hơn hàm Manhattan Norm** cho trường hợp của thuật toán A*, còn đối với thuật toán GBFS cả 2 hàm heuristic đều cho ra đường đi tối ưu và không duyệt dư điểm nào.

Bản đồ 5:

Bản đồ không có điểm thưởng.

Điểm xuất phát đặt tại $(9, 5)$.

Điểm đích đặt tại $(10, 15)$.

Thuật toán Depth First Search (DFS):

Đường đi tìm được có chi phí là **41**.

Thuật toán Breadth First Search (BFS):

A 20x20 grid world environment. The start state is a yellow star at (19, 10). The goal state is a red triangle at (19, 18). Obstacles are black 'X' marks forming a complex shape. The word 'EXIT' is written in red at the bottom right.

Đường đi tìm được có chi phí là **17**.

Thuật toán Greedy Best First Search (GBFS):

Sử dụng Euclid Norm.

```

XXXXXXXXXXXXXXXXXXXXXXXXX
X                                  X
X      X  XXXXX  XX  X  XXXX
X  X      XXX  XXXX  XX  XX
X  X      X  X      X      X
X                                  XX
XX  XX      >>>>  XX  XX
X      XXXXX  ▲>>>>  XX  XX
X      >>>>>>>>▲>>>>  XX
XXXXXXXX★XX  X  X  X  X  X  X
XXXXXXXXXXXXXXXXXXXXXXXXEXIT

```

Đường đi tìm được có chi phí là 17.

Sử dụng Manhattan Norm.

```

XXXXXXXXXXXXXXXXXXXXXXXXX
X                                  X
X      X  XXXXX  XX  X  XXXX
X  X      XXXX  XXXX  XX  XX
X  X      X  X      X      X
X                                  XX
XX  XX      >>>>>>>>XX  XX
X      XXXXX  ●▲>>>>  XX  XX
X      >>>>>>>>▲>>>>  XX
XXXXXXXX★XX  ●●●●●  X  X  X  X
XXXXXXXXXXXXXXXXXXXXXXXXEXIT

```

Đường đi tìm được có chi phí là 17.

Thuật toán A*:

Sử dụng Euclid Norm.

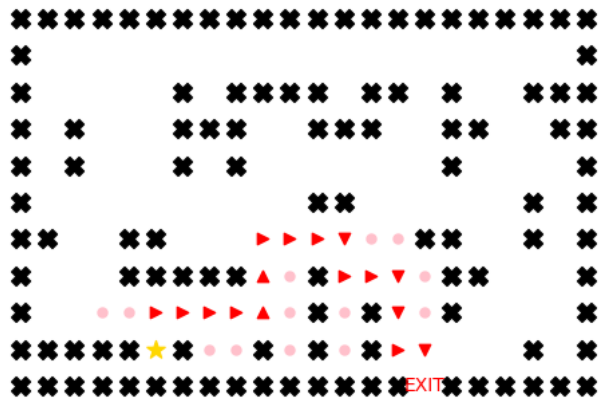
```

XXXXXXXXXXXXXXXXXXXXXXXXX
X                                  X
X      X  XXXXX  XX  X  XXXX
X  X      XXXX  XXXX  XX  XX
X  X      X  X      X      X
X                                  XX
XX  XX      ●●XX      XX  XX
XX  XX      ●>>>>>>>>>XX  XX
X      ●XXXXXX▲>>>>●>>>>XX  XX
X      ●>>>>>>>>▲>>>>XX  XX
XXXXXXXX★XX  ●●●●●  X  X  X  X
XXXXXXXXXXXXXXXXXXXXXXXXEXIT

```

Đường đi tìm được có chi phí là 17.

Sử dụng Manhattan Norm.



Đường đi tìm được có chi phí là **17**.

Nhận xét:

- Đối với bản đồ thứ 5, thuật toán GBFS sử dụng cả Euclid Norm là thuật toán tìm được đường đi tốt nhất, với chi phí đường đi là **17 và chỉ duyệt duy nhất 1 điểm**.
 - A* cũng cho ra đường đi tối ưu. Vì thuật toán này là thuật toán vừa hoàn thiện vừa tối ưu nên sẽ **đảm bảo tìm được đường đi tối ưu** (nếu có).
 - Thuật toán tìm kiếm mù là BFS cũng cho ra đường đi tối ưu với chi phí là **27** vì BFS là thuật toán tìm ra đường đi tối ưu nếu chi phí ở mỗi điểm là như nhau. Tuy nhiên, số lượng điểm mà BFS duyệt qua lớn và cũng lớn nhất trong các thuật toán này (ở trường hợp này BFS **duyệt khoảng 2/3 số điểm** trên bản đồ).
 - Đối với thuật toán DFS đường đi tìm được có chi phí **41** lớn gấp hơn 2 lần so với đường đi tối ưu. Điều này cũng không có gì đáng ngạc nhiên vì DFS được đánh giá là thuật toán không tối ưu.
 - Ở bản đồ thứ 5 thì đối với 2 thuật toán tìm kiếm có thông tin thì **hàm Euclid Norm có vẻ hoạt động tốt hơn hàm Manhattan Norm** cho trường hợp của thuật toán GBFS, còn đối với thuật toán A* thì hàm Manhattan Norm có vẻ hoạt động tốt hơn hàm Manhattan Norm. Tuy nhiên sự chênh lệch là khá ít giữa 2 hàm Heuristic, chỉ duyệt hơn khoảng 5 điểm.
- Đánh giá chung các thuật toán trên bản đồ không có điểm thưởng.
- Thuật toán Depth First search (DFS).
- Là thuật toán không tối ưu, thường cho kết quả là đường đi có chi phí lớn hơn so với các thuật toán còn lại.
 - Không có một xu hướng duyệt nào cụ thể, cứ duyệt cho đến khi cụt đường và quay lại duyệt tiếp các đỉnh khác nên đường đi rất khó có thể dự đoán được.
 - Là thuật toán có số điểm được duyệt qua tương đối lớn, chỉ đứng sau thuật toán BFS.

- Việc tìm được đường đi tối ưu hay không có phụ thuộc vào thứ tự các điểm được push vào và pop ra (do Lập trình viên quyết định). Đối với đoạn mã này, thứ tự push vào là trên, phải, dưới, trái và do đó thứ tự được pop ra tương ứng là trái, dưới, phải, trên.

Thuật toán Breadth First Search (BFS).

- Là thuật toán tối ưu nếu chi phí đường đi tại mọi điểm là như nhau. Và trong trường hợp này, BFS luôn cho kết quả là đường đi tối ưu nhất.
- Có xu hướng mở rộng đường đi đều ra tất cả các phía với các mức 1 bước đi, 2 bước đi, k bước đi, cho đến đích.
- Số điểm được duyệt qua là rất lớn và lớn nhất trong các thuật toán này. Nhiều trường hợp BFS duyệt qua hết cả bản đồ.

Thuật toán Greedy Best First Search (GBFS).

- Không phải là thuật toán có thể đảm bảo tìm được đường đi tối ưu nhất.
- Có xu hướng mở rộng đường đi hướng tới các điểm có chi phí ước lượng gần đích nhất.
- Số điểm duyệt qua tương đối ít, là ít nhất trong các thuật toán này.
- Việc tìm được đường đi tối ưu hay không phụ thuộc vào hàm Heuristic và bản đồ.

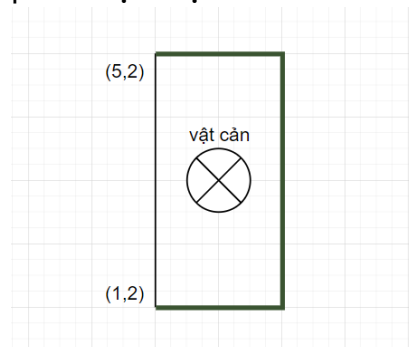
Thuật toán A*.

- Là thuật toán đảm bảo tối ưu nếu hàm Heuristic hợp lý và nhất quán.
- Có xu hướng mở rộng điểm hướng tới đích, không mở rộng đều như BFS.
- Số điểm duyệt qua khá nhiều đối với thuật toán tìm kiếm có thông tin, lớn hơn nhiều lần so với GBFS.
- Trong trường hợp chi phí mọi điểm là như nhau và bằng 1 này Euclid Norm và Hammett Norm đều là 2 hàm Heuristic hợp lý, nhất quán nên A* luôn đảm bảo được đường đi tối ưu nhất.

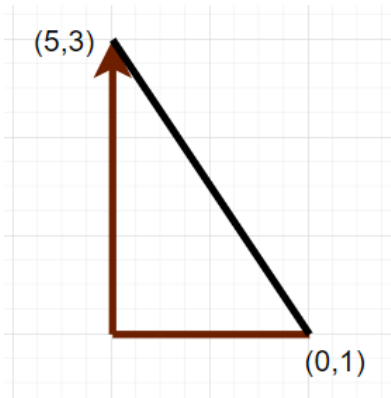
Đánh giá chung 2 hàm Heuristic trên bản đồ không có điểm thưởng.
Hàm Euclid Norm.

- Chi phí ước lượng là hợp lý vì:

Nếu đường thẳng đi qua 1 điểm và đích song song với 1 trục nào đó, chi phí thực sự luôn lớn hơn hoặc bằng chi phí ước tính (nếu có đường đi).



Nếu đường thẳng đi qua 1 điểm và đích cắt 2 trục, chi phí thực sự luôn lớn hơn chi phí ước tính (nếu có đường đi).



- Số điểm được duyệt qua lớn hơn so với hàm Manhattan Norm.

Hàm Manhattan Norm.

- Chi phí ước lượng là hợp lý vì khoảng cách dx , dy hầu hết đều bằng số bước dịch chuyển lên, xuống, trái, phải và trong một số trường hợp xấu nhiều vật cản thì chi phí di chuyển thực sẽ lớn hơn tổng dx và dy .

- Số điểm duyệt qua ít hơn so với hàm Euclid Norm.

Có thể nói rằng trong trường hợp bản đồ không có điểm thưởng, hàm Heuristic Manhattan là tốt hơn so với hàm Euclid Norm.

So sánh các thuật toán trên bản đồ có điểm thưởng.

Phát biểu các thuật toán.

Đối với thuật toán thứ I (Search T): dựa trên thuật toán A_t

- Có cấu trúc gần giống với thuật toán A_t .
- Không update các điểm đã duyệt qua rồi trong tập $close$.
- Có $f(x)=g(n)$. (với $g(n)$ là chi phí đường đi)
- Mở các điểm có chi phí đường đi thấp nhất mà không nằm trong tập $close$ và tập $open$.

⇒ Vì để tránh thuật toán đi vào vòng lặp vô tận (chu trình âm) nên ta bỏ qua tập $close$ không update chi phí cho các điểm đã duyệt qua rồi.

Phần còn lại của thuật toán tương tự thuật toán A_t .

Đối với thuật toán thứ II (search*): dựa trên thuật toán A^*

- Có cấu trúc thuật toán gần giống với thuật toán A^*
- Không update các điểm đã duyệt qua rồi trong tập $close$.
- Có $f(x)=h(x)+g(n)$. (với $g(n)$ là chi phí đường đi và $h(x)$ là chi phí đường đi ước tính)
- Mở các điểm có chi phí đường đi + chi phí ước tính đến đích là nhỏ nhất mà không nằm trong tập $close$ và tập $open$.

⇒ Vì để tránh thuật toán đi vào vòng lặp vô tận (chu trình âm) nên ta bỏ qua tập $close$ không update chi phí cho các điểm đã duyệt qua rồi.

Phần còn lại của thuật toán tương tự thuật toán A^* .

Bản đồ 1:



Có điểm thưởng đặt tại các điểm cộng trên bản đồ.
Điểm xuất phát đặt tại (11, 28).
Và điểm thoát đặt tại (4, 0).

Thuật toán Search T:



Đường đi tìm được có chi phí là 33.

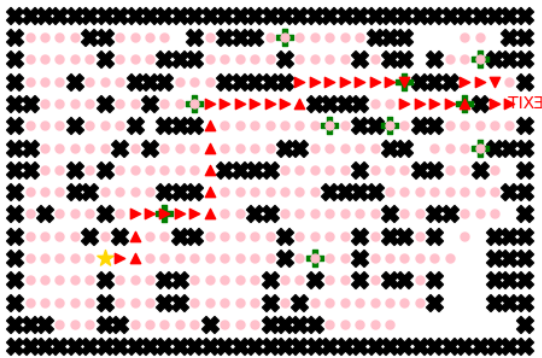
Thuật toán Search *:

Sử dụng Euclid Norm:



Đường đi tìm được có chi phí là 33.

Sử dụng Manhattan Norm:



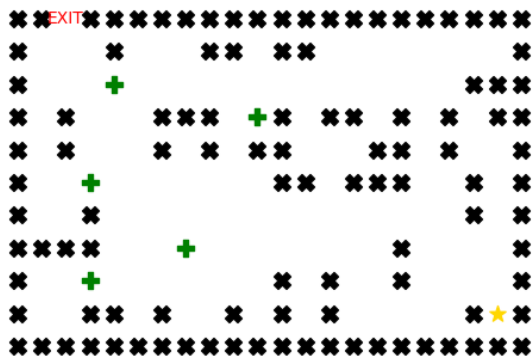
Đường đi tìm được có chi phí là 33.

Nhận xét:

- đối với bản đồ 1 thì thuật toán Search * sử dụng hàm heuristic Là Manhattan Norm là thuật toán tốt nhất khi tìm ra được đường đi tối ưu nhất với số điểm duyệt qua cũng ít nhất trong các thuật toán.
- riêng thuật toán Search* sử dụng hàm heuristic Là Euclid Norm là thuật toán không tối ưu vì chỉ tìm ra được đường đi tối ưu chứ không tối ưu nhất với chi phí đường đi là 36 bên cạnh đó số điểm duyệt qua cũng khá lớn (gần như bao phủ cả bản đồ).

⇒ Đối với thuật toán Search* trong bản đồ đầu tiên thì hàm Manhattan Norm có vẻ hoạt động tốt hơn hàm Euclid Norm khi cho ra được đường đi tối ưu. Bên cạnh đó thuật toán Search T cũng cho ra được đường đi tối ưu nhưng số điểm duyệt qua lại lớn hơn so với thuật toán Search*.

Bản đồ 2:



Bản đồ có 5 điểm thưởng được đại diện bởi các dấu + trên bản đồ.

Điểm xuất phát đặt tại (9, 20).

Điểm đích đặt tại (0, 2).

Thuật toán Search T:

Đường đi tìm được có chi phí là **24**.

Thuật toán Search*:

Sử dụng Euclid Norm.

Đường đi tìm được có chi phí là **25**.

Sử dụng Manhattan Norm.

Đường đi tìm được có chi phí là **25**.

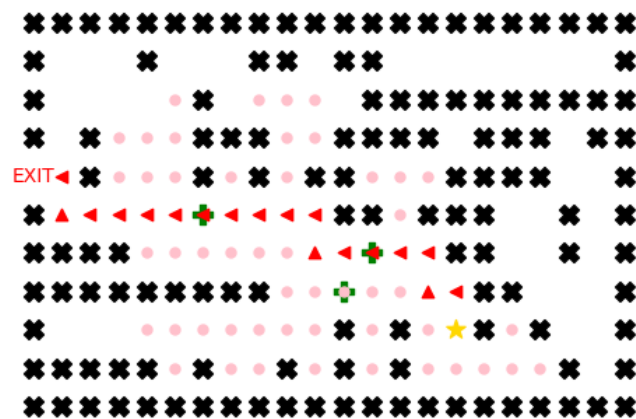
Nhận xét:

- đối với bản đồ 2 thì thuật toán Search T là thuật toán tốt nhất khi tìm ra được đường đi tối ưu nhất, tuy nhiên số điểm duyệt qua gần như hết cả bản đồ.
- thuật toán Search * sử dụng hàm heuristic là cả Euclid Norm và Hanmattan tìm ra đường đi không tối ưu với chi phí đường đi là 25 tuy nhiên số điểm duyệt qua có phần nhỉnh hơn so với search T.

Bản đồ 3:

Đường đi tìm được có chi phí là **15**.

Sử dụng Manhattan Norm.



Đường đi tìm được có chi phí là 14.

Nhận xét:

- đối với bản đồ 3 thì thuật toán Search * sử dụng hàm heuristic Là Manhattan Norm là thuật toán tốt nhất khi tìm ra được đường đi tối ưu nhất với số điểm duyệt qua cũng ít nhất trong các thuật toán.
 - riêng thuật toán Search * sử dụng hàm heuristic Là Euclid Norm không tìm ra được đường đi tối ưu với chi phí đường đi là 15 bên cạnh đó số điểm duyệt qua cũng lớn hơn so với Search T và Search * sử dụng Manhattan.
- ⇒ Đối với thuật toán Search* trong bản đồ 3 thì hàm Manhattan Norm có vẻ hoạt động tốt hơn hàm Euclid Norm khi cho ra được đường đi có chi phí thấp hơn. Bên cạnh đó thuật toán Search T cũng cho ra được đường đi có cùng chi phí với Manhattan Norm nhưng số điểm duyệt qua lại lớn hơn so với thuật toán Search*.

Đánh giá chung các thuật toán trên bản đồ có điểm thưởng.

Thuật toán Search T:

- Là thuật toán tìm kiếm mù không đảm bảo cho ra được đường đi tối ưu.
- Có xu hướng duyệt hướng đến chỉ một vài điểm thưởng (vì không update tập close) => có thể bỏ qua một số đường đi có chi phí thấp hơn bao gồm có điểm thưởng.
- Việc tìm được đường đi tối ưu hay không phụ thuộc vào chi phí tại mỗi điểm và loại bản đồ.

Thuật toán Search *:

- Là thuật toán tìm kiếm có thông tin không đảm bảo cho ra được đường đi tối ưu vì:

Tồn tại chi phí tại một điểm mang giá trị âm, và để tránh xảy ra lặp vòng lặp vô tận vì có thể tồn tại chu trình âm nên đã bỏ qua việc update các điểm có trong close => có thể bỏ qua các đường đi có chi phí thấp hơn.

Cả 2 hàm Heuristic Euclid Norm và Manhattan Norm đều không hợp lý vì giá trị thực của đường đi có thể nhỏ hơn giá trị đường đi ước tính.

- Có xu hướng duyệt hướng đến chỉ một vài điểm thưởng và cũng hướng đến điểm đích.

- Việc có thể tìm ra đường đi tối ưu hay không phần lớn phụ thuộc vào loại bản đồ, chi phí khác nhau tại các điểm và cả hàm heuristic được sử dụng.

Dựa vào 3 bản đồ trên thì Search * sử dụng Manhattan Norm có phần tốt hơn đôi chút so với sử dụng Euclid Norm. Tuy nhiên, cả 2 hàm Heuristic này cũng khá tương đương nhau và ta cần có thêm nhiều loại bản đồ khác để có thể khảo sát thêm về việc đánh giá hai hàm này.

Link demo: <https://youtu.be/YnhuyJ9-rpQ>