

BÁO CÁO ĐỒ ÁN 1:

ỨNG DỤNG CÁC GIẢI THUẬT TÌM KIẾM



Nhóm gồm có:

Nguyễn Văn Thịnh – 19120668

Phan Văn Thắng – 19120656

Mục lục

Tóm tắt cách giải quyết bài toán	3
I. Bản đồ không có điểm thưởng	3
1. Bản đồ bình thường (không phức tạp).....	3
2. Bản đồ có khoảng cách từ điểm bắt đầu gần điểm kết thúc (theo đường chim bay)	7
3. Bản đồ trống (không có vật cản)	12
4. Bản đồ nhiều vật cản chỉ có 1 đường thoát duy nhất	17
5. Bản đồ lớn phức tạp, chật hẹp với nhiều vật cản.....	21
Nhận xét các thuật toán tìm đường đi trong mê cung	25
II. Bản đồ có điểm thưởng	26
1. Bản đồ bình thường (không phức tạp) với 2 điểm thưởng	26
2. Bản đồ có khoảng cách từ điểm bắt đầu gần điểm kết thúc (theo đường chim bay) với 5 điểm thưởng	28
3. Bản đồ lớn phức tạp, chật hẹp với nhiều vật cản với 10 điểm thưởng	30
Nhận xét:	34

Tóm tắt cách giải quyết bài toán bằng các thuật toán và hàm heuristic.

Thuật toán	Mô tả
BFS	Cố gắng mở tất cả các điểm con của một điểm đang xét.
DFS	Mở điểm con của điểm đang xét rồi tiếp tục mở điểm con của điểm đó đến khi không thể mở được nữa.
Greedy-Best First Search	Cố gắng mở điểm con được ước lượng khoảng cách gần đích nhất $h(x)$.
A*	Mở điểm con có ước lượng khoảng cách gần điểm bắt đầu nhất $g(x)$ và có ước lượng khoảng cách gần đích nhất $h(x)$. Hàm $f(x) = g(x) + h(x)$ có giá trị càng thấp thì độ ưu tiên càng cao.

■ Đối với thuật toán Greedy-BFS và A*:

Sử dụng 2 hàm heuristic: Tính khoảng cách bằng Euclid (tính bằng độ dài của đường thẳng nối 2 điểm) và Manhattan (tính bằng tổng chiều dài của hình chiếu của đường thẳng nối 2 điểm)

I. Bản đồ không có điểm thưởng

1. Bản đồ bình thường (không phức tạp)

The height of the matrix: 15

The width of the matrix: 25

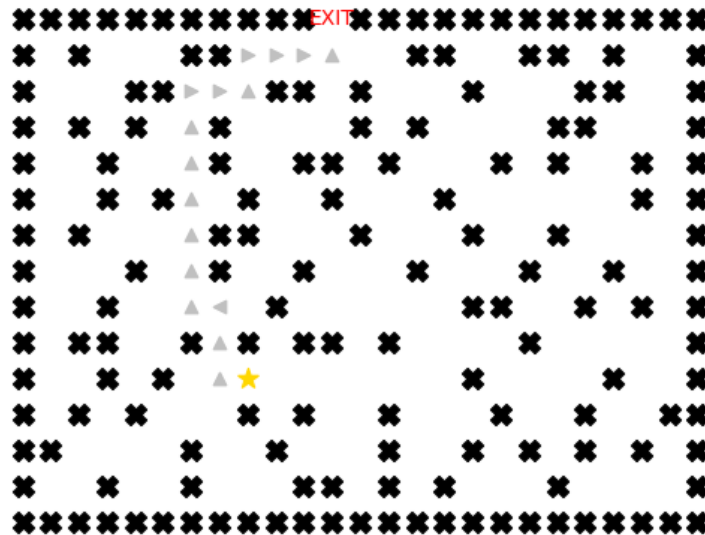
Starting point $(x, y) = (10, 8)$ -> Ending point $(x, y) = (0, 11)$

a) Thuật toán tìm kiếm không có thông tin:

+ Thuật toán tìm kiếm DFS (Depth First Search)

► Chi phí: 17

► Path: [(10, 8), (10, 7), (9, 7), (8, 7), (8, 6), (7, 6), (6, 6), (5, 6), (4, 6), (3, 6), (2, 6), (2, 7), (2, 8), (1, 8), (1, 9), (1, 10), (1, 11), (0, 11)]

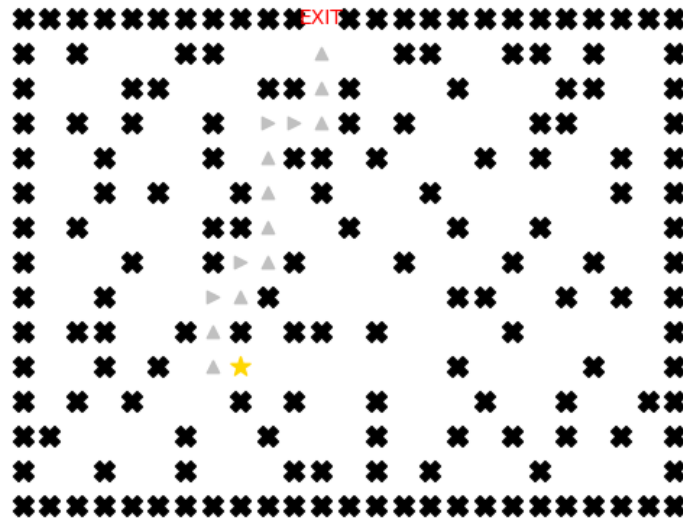


► Tìm được đường đi ra ngoài nhưng chi phí không tối ưu, số nút mở: 30

+ Thuật toán tìm kiếm BFS (Breadth First Search)

► Chi phí: 15

► Path: [(10, 8), (10, 7), (9, 7), (8, 7), (8, 8), (7, 8), (7, 9), (6, 9), (5, 9), (4, 9), (3, 9), (3, 10), (3, 11), (2, 11), (1, 11), (0, 11)]



- Tìm được đường đi ra ngoài với chi phí tối ưu, số nút mở: 354

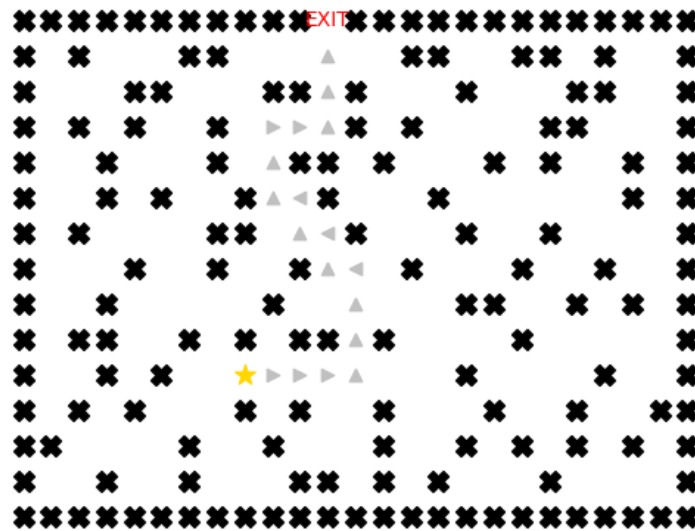
b) Thuật toán tìm kiếm có thông tin:

- Hàm heuristic đánh giá bằng khoảng cách Euclid

+ Thuật toán tìm kiếm tham lam (Greedy Best First Search)

- Chi phí: 19

- Path: [(10, 8), (10, 9), (10, 10), (10, 11), (10, 12), (9, 12), (8, 12), (7, 12), (7, 11), (6, 11), (6, 10), (5, 10), (5, 9), (4, 9), (3, 9), (3, 10), (3, 11), (2, 11), (1, 11), (0, 11)]

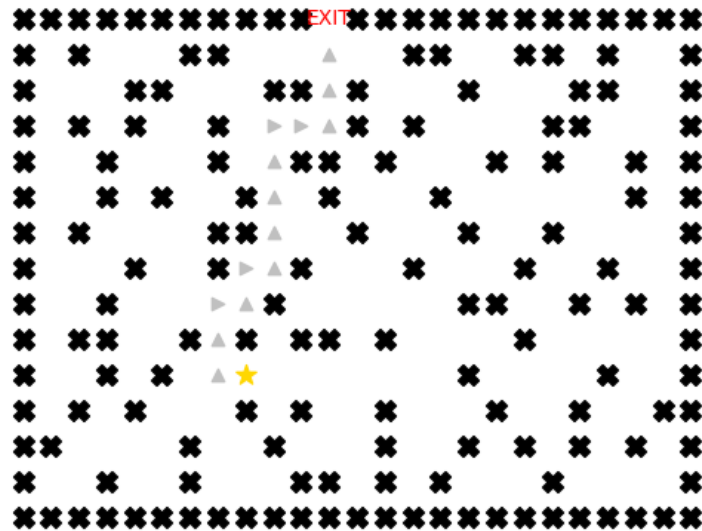


► Tìm được đường đi ra ngoài nhưng chi phí không tối ưu, chi phí bỏ ra cao hơn
DFS, số nút mở: 35

+ Thuật toán tìm kiếm A*

► Chi phí: 15

► Path: [(10, 8), (10, 7), (9, 7), (8, 7), (8, 8), (7, 8), (7, 9), (6, 9), (5, 9), (4, 9), (3, 9), (3, 10), (3, 11), (2, 11), (1, 11), (0, 11)]



-> Tìm được đường đi ra ngoài với chi phí tối ưu như BFS, số nút mở 31

■ Hàm Heuristic đánh giá bằng khoảng cách Manhattan

+ Thuật toán tìm kiếm tham lam (Greedy Best First Search)

► Cách đi khác nhưng cùng chi phí với hàm heuristic Euclid, số nút mở: 35

+ Thuật toán tìm kiếm A*

► Cùng chi phí với hàm heuristic Euclid, số nút mở: 55

★ Kết luận về bản đồ 1:

Chi phí: $BFS = A^* < DFS < Greedy$

Số nút mở: $DFS < A^*(Euclid) < Greedy < A^*(Manhattan) < BFS$

2. Bản đồ có khoảng cách từ điểm bắt đầu gần điểm kết thúc (theo đường chim bay)

The height of the matrix: 15

The width of the matrix: 25

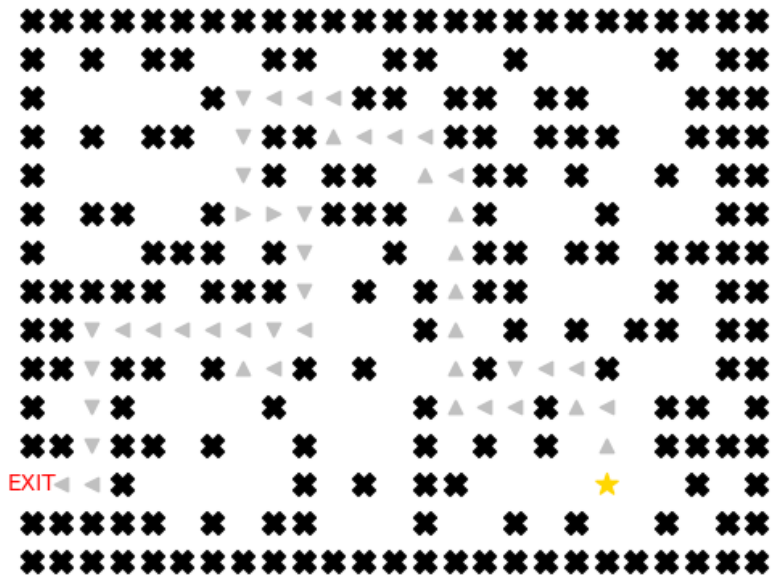
Starting point (x, y) = (12, 19) -> Ending point (x, y) = (12, 0)

a) Thuật toán tìm kiếm không có thông tin:

+ Thuật toán tìm kiếm DFS

► Chi phí: 47

► Path: [(12, 19), (11, 19), (10, 19), (10, 18), (9, 18), (9, 17), (9, 16), (10, 16), (10, 15), (10, 14), (9, 14), (8, 14), (7, 14), (6, 14), (5, 14), (4, 14), (4, 13), (3, 13), (3, 12), (3, 11), (3, 10), (2, 10), (2, 9), (2, 8), (2, 7), (3, 7), (4, 7), (5, 7), (5, 8), (5, 9), (6, 9), (7, 9), (8, 9), (8, 8), (9, 8), (9, 7), (8, 7), (8, 6), (8, 5), (8, 4), (8, 3), (8, 2), (9, 2), (10, 2), (11, 2), (12, 2), (12, 1), (12, 0)]

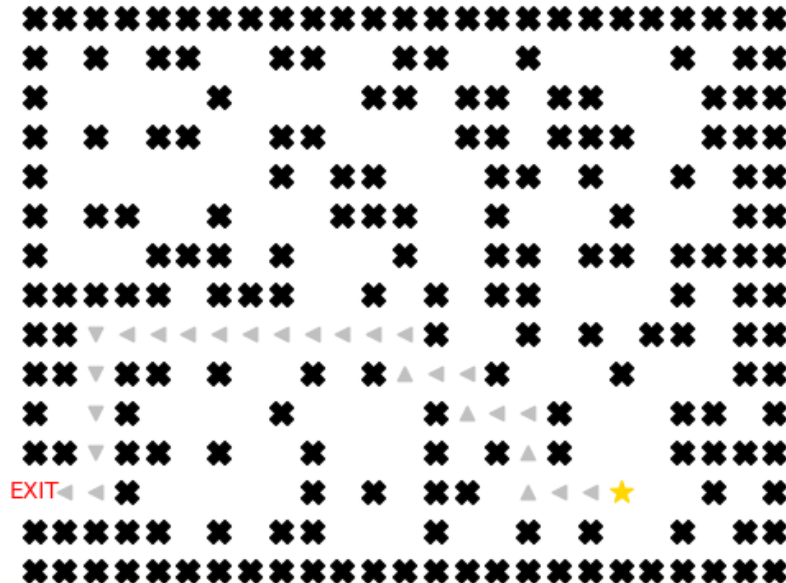


► Tìm được đường đi ra ngoài nhưng chi phí không tối ưu, số nút mở: 133

+ Thuật toán tìm kiếm BFS

► Chi phí: 27

► Path: [(12, 19), (12, 18), (12, 17), (12, 16), (11, 16), (10, 16), (10, 15), (10, 14), (9, 14), (9, 13), (9, 12), (8, 12), (8, 11), (8, 10), (8, 9), (8, 8), (8, 7), (8, 6), (8, 5), (8, 4), (8, 3), (8, 2), (9, 2), (10, 2), (11, 2), (12, 2), (12, 1), (12, 0)]



► Tìm được đường đi ra ngoài với chi phí tốt hơn DFS, số nút mở: 417

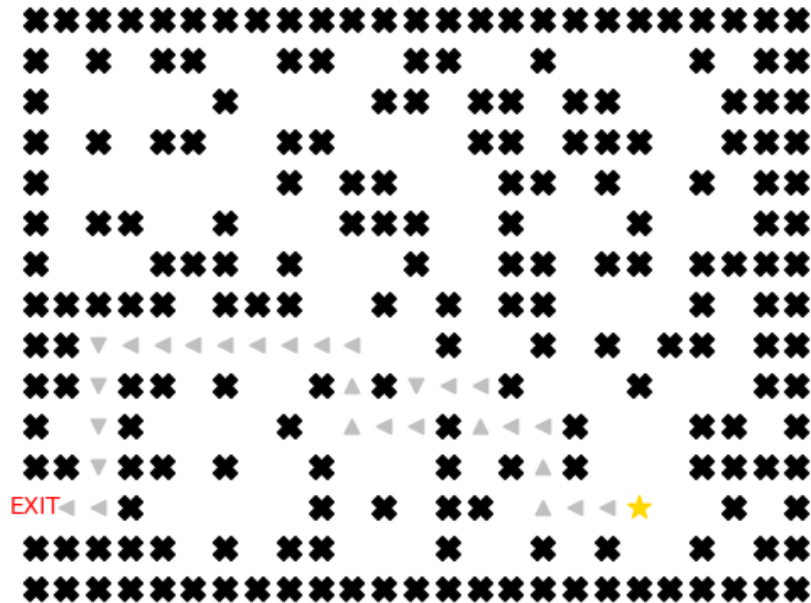
b) Thuật toán tìm kiếm có thông tin:

■ Hàm heuristic đánh giá bằng khoảng cách Euclid

+ Thuật toán tìm kiếm tham lam

Chi phí: 29

Path: [(12, 19), (12, 18), (12, 17), (12, 16), (11, 16), (10, 16), (10, 15), (10, 14), (9, 14), (9, 13), (9, 12), (10, 12), (10, 11), (10, 10), (9, 10), (8, 10), (8, 9), (8, 8), (8, 7), (8, 6), (8, 5), (8, 4), (8, 3), (8, 2), (9, 2), (10, 2), (11, 2), (12, 2), (12, 1), (12, 0)]

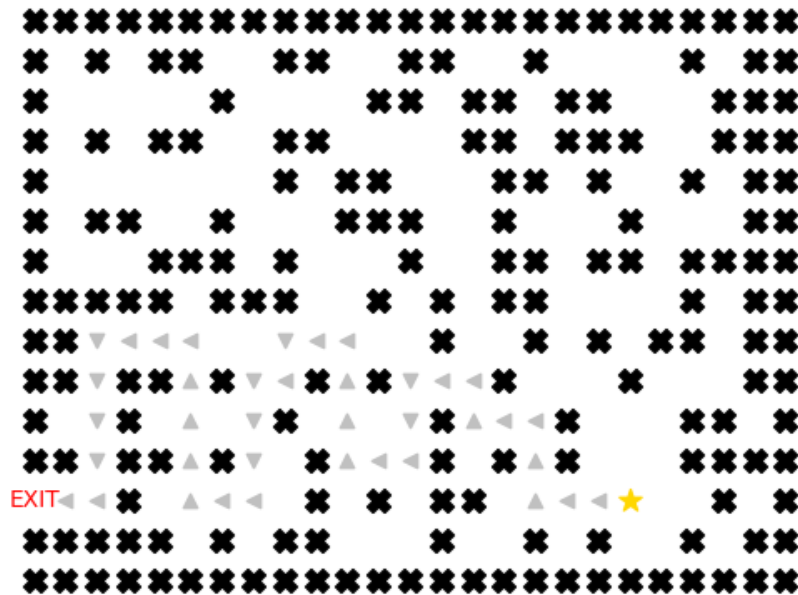


- Tìm được đường đi ra nhưng chi phí không tối ưu bằng BFS, số nút mở: 57

+ Thuật toán tìm kiếm A*

- Chi phí: 39

► Path: [(12, 19), (12, 18), (12, 17), (12, 16), (11, 16), (10, 16), (10, 15), (10, 14), (9, 14), (9, 13), (9, 12), (10, 12), (11, 12), (11, 11), (11, 10), (10, 10), (9, 10), (8, 10), (8, 9), (8, 8), (9, 8), (9, 7), (10, 7), (11, 7), (12, 7), (12, 6), (12, 5), (11, 5), (10, 5), (9, 5), (8, 5), (8, 4), (8, 3), (8, 2), (9, 2), (10, 2), (11, 2), (12, 2), (12, 1), (12, 0)]



- Tìm được đường đi ra ngoài nhưng chi phí không tối ưu bằng Greedy, số nút mở: 127

■ Hàm Heuristic đánh giá bằng khoảng cách Manhattan

+ Thuật toán tìm kiếm tham lam

- Giống trong hàm heuristic Euclid, số nút mở: 62

+ Thuật toán tìm kiếm A*

- Giống trong hàm heuristic Euclid, số nút mở: 147

★ Kết luận về bản đồ 2:

Chi phí: BFS < Greedy < A* < DFS

Số nút mở:

Greedy(Euclid) < Greedy(Manhattan) < A*(Euclid) < DFS < A*(Manhattan) < BFS

3. Bản đồ trống (không có vật cản)

The height of the matrix: 15

The width of the matrix: 25

Starting point (x, y) = (7, 12) -> Ending point (x, y) = (1, 0)

a) Thuật toán tìm kiếm không có thông tin:

+ Thuật toán tìm kiếm DFS

► Chi phí: 138

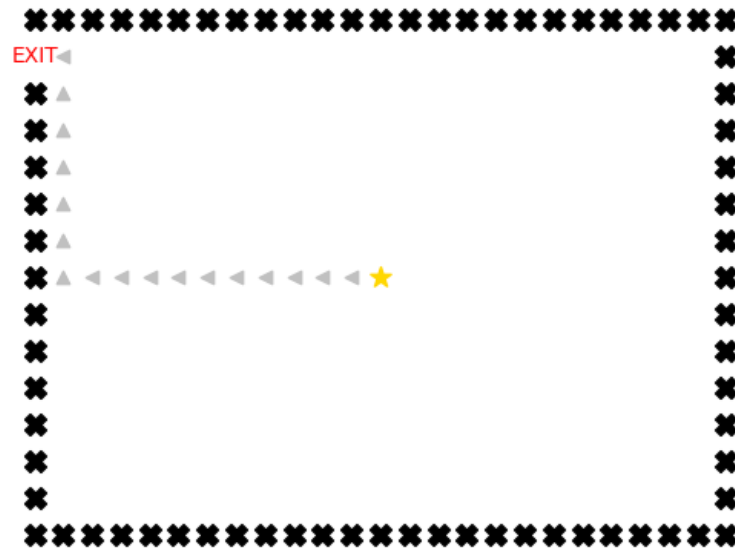
► Path: [(7, 12), (6, 12), (5, 12), (4, 12), (3, 12), (2, 12), (1, 12), (1, 11), (2, 11), (3, 11), (4, 11), (5, 11), (6, 11), (7, 11), (8, 11), (9, 11), (10, 11), (11, 11), (12, 11), (13, 11), (13, 10), (12, 10), (11, 10), (10, 10), (9, 10), (8, 10), (7, 10), (6, 10), (5, 10), (4, 10), (3, 10), (2, 10), (1, 10), (1, 9), (2, 9), (3, 9), (4, 9), (5, 9), (6, 9), (7, 9), (8, 9), (9, 9), (10, 9), (11, 9), (12, 9), (13, 9), (13, 8), (12, 8), (11, 8), (10, 8), (9, 8), (8, 8), (7, 8), (6, 8), (5, 8), (4, 8), (3, 8), (2, 8), (1, 8), (1, 7), (2, 7), (3, 7), (4, 7), (5, 7), (6, 7), (7, 7), (8, 7), (9, 7), (10, 7), (11, 7), (12, 7), (13, 7), (13, 6), (12, 6), (11, 6), (10, 6), (9, 6), (8, 6), (7, 6), (6, 6), (5, 6), (4, 6), (3, 6), (2, 6), (1, 6), (1, 5), (2, 5), (3, 5), (4, 5), (5, 5), (6, 5), (7, 5), (8, 5), (9, 5), (10, 5), (11, 5), (12, 5), (13, 5), (13, 4), (12, 4), (11, 4), (10, 4), (9, 4), (8, 4), (7, 4), (6, 4), (5, 4), (4, 4), (3, 4), (2, 4), (1, 4), (1, 3), (2, 3), (3, 3), (4, 3), (5, 3), (6, 3), (7, 3), (8, 3), (9, 3), (10, 3), (11, 3), (12, 3), (13, 3), (13, 2), (12, 2), (11, 2), (10, 2), (9, 2), (8, 2), (7, 2), (6, 2), (5, 2), (4, 2), (3, 2), (2, 2), (1, 2), (1, 1), (1, 0)]



- Tìm được đường đi ra ngoài nhưng chi phí quá tốn kém, số nút mở: 290

+ Thuật toán tìm kiếm BFS

- Chi phí: 18
- Path: [(7, 12), (7, 11), (7, 10), (7, 9), (7, 8), (7, 7), (7, 6), (7, 5), (7, 4), (7, 3), (7, 2), (7, 1), (6, 1), (5, 1), (4, 1), (3, 1), (2, 1), (1, 1), (1, 0)]



- Tìm được đường đi ra ngoài với chi phí tối ưu, số nút mở: 213886

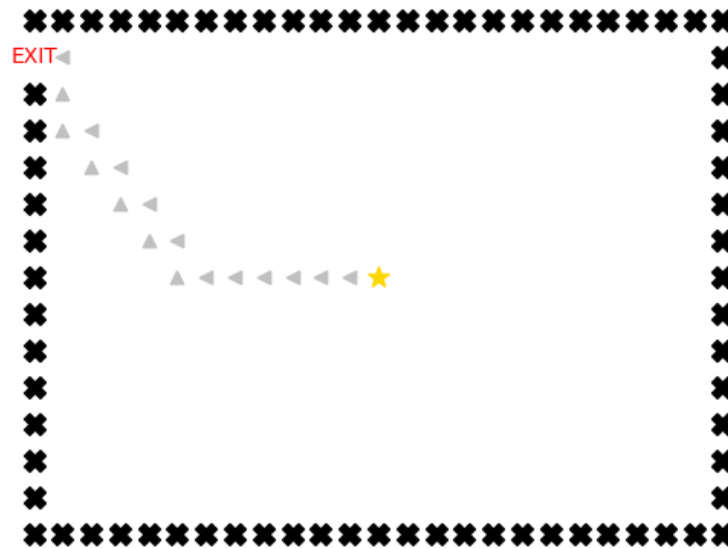
b) Thuật toán tìm kiếm có thông tin:

■ Hàm heuristic đánh giá bằng khoảng cách Euclid

+ Thuật toán tìm kiếm tham lam

- Chi phí: 18

► Path: [(7, 12), (7, 11), (7, 10), (7, 9), (7, 8), (7, 7), (7, 6), (7, 5), (6, 5), (6, 4), (5, 4), (5, 3), (4, 3), (4, 2), (3, 2), (3, 1), (2, 1), (1, 1), (1, 0)]



► Tìm được đường đi ra ngoài với chi phí tối ưu (nhưng cách đi khác với BFS), số nút mở: 52

+ Thuật toán tìm kiếm A*

► Cost: 18

► Path: [(7, 12), (7, 11), (6, 11), (6, 10), (6, 9), (5, 9), (5, 8), (5, 7), (4, 7), (4, 6), (4, 5), (3, 5), (3, 4), (3, 3), (2, 3), (2, 2), (2, 1), (1, 1), (1, 0)]



4. Bản đồ nhiều vật cản chỉ có 1 đường thoát duy nhất

The height of the matrix: 15

The width of the matrix: 25

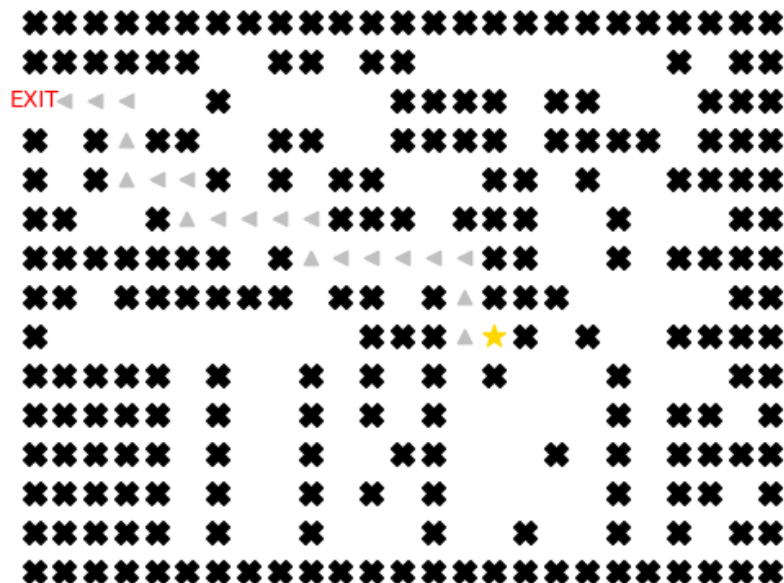
Starting point (x, y) = (8, 15) -> Ending point (x, y) = (2, 0)

a) Thuật toán tìm kiếm không có thông tin:

+ Thuật toán tìm kiếm DFS

► Chi phí: 21

► Path: [(8, 15), (8, 14), (7, 14), (6, 14), (6, 13), (6, 12), (6, 11), (6, 10), (6, 9), (5, 9), (5, 8), (5, 7), (5, 6), (5, 5), (4, 5), (4, 4), (4, 3), (3, 3), (2, 3), (2, 2), (2, 1), (2, 0)]

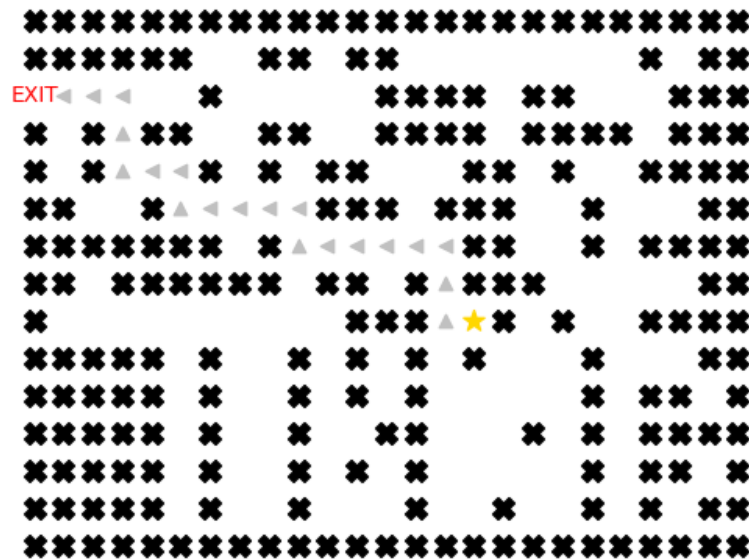


► Tìm được đường đi thoát ra ngoài, số nút mở: 48

+ Thuật toán tìm kiếm BFS

► Chi phí: 21

► Path: [(8, 15), (8, 14), (7, 14), (6, 14), (6, 13), (6, 12), (6, 11), (6, 10), (6, 9), (5, 9), (5, 8), (5, 7), (5, 6), (5, 5), (4, 5), (4, 4), (4, 3), (3, 3), (2, 3), (2, 2), (2, 1), (2, 0)]



- Tìm được đường đi thoát ra ngoài, số nút mở: 166

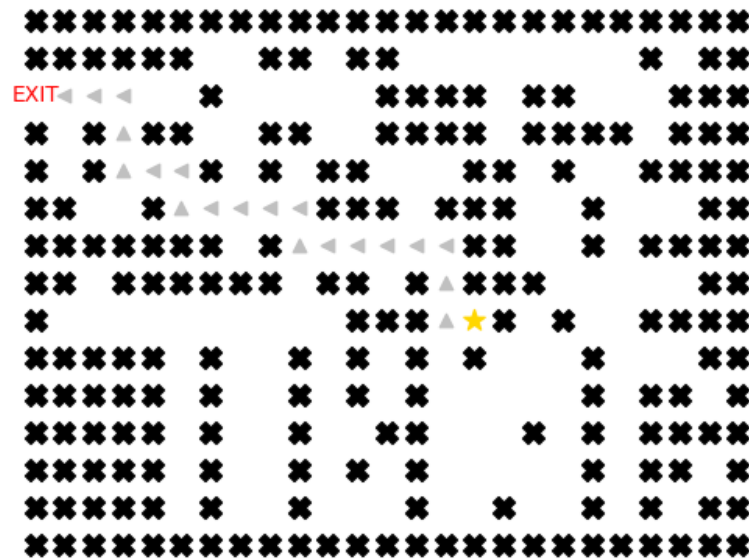
b) Thuật toán tìm kiếm có thông tin:

■ Hàm heuristic đánh giá bằng khoảng cách Euclid

+ Thuật toán tìm kiếm tham lam

- Chi phí: 21

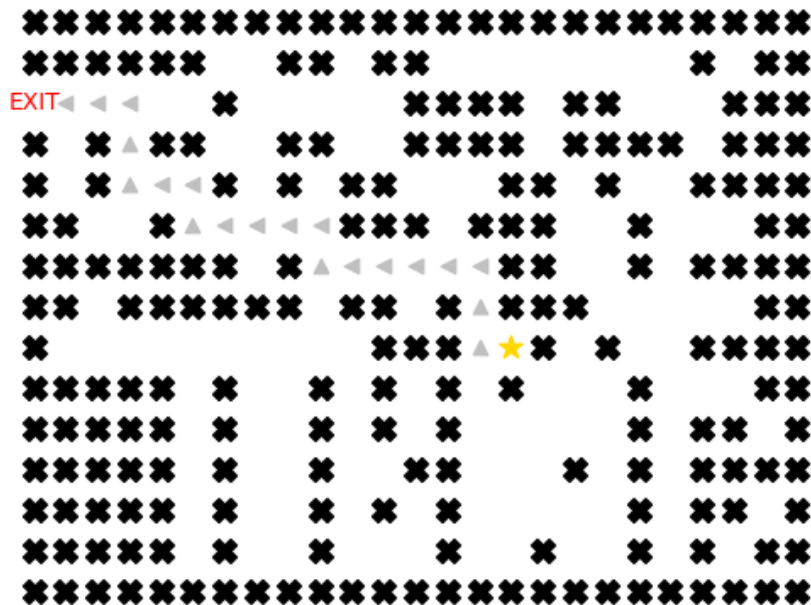
► Path: [(8, 15), (8, 14), (7, 14), (6, 14), (6, 13), (6, 12), (6, 11), (6, 10), (6, 9), (5, 9), (5, 8), (5, 7), (5, 6), (5, 5), (4, 5), (4, 4), (4, 3), (3, 3), (2, 3), (2, 2), (2, 1), (2, 0)]



- Tìm được đường đi thoát ra ngoài, số nút mở: 31

+ Thuật toán tìm kiếm A*

- Chi phí: 21
- Path: [(8, 15), (8, 14), (7, 14), (6, 14), (6, 13), (6, 12), (6, 11), (6, 10), (6, 9), (5, 9), (5, 8), (5, 7), (5, 6), (5, 5), (4, 5), (4, 4), (4, 3), (3, 3), (2, 3), (2, 2), (2, 1), (2, 0)]



- Tìm được đường đi thoát ra ngoài, số nút mở: 33

■ Hàm Heuristic đánh giá bằng khoảng cách Manhattan

+ Thuật toán tìm kiếm tham lam

- Giống trong hàm heuristic Euclid, số nút mở: 31

+ Thuật toán tìm kiếm A*

- Giống trong hàm heuristic Euclid, số nút mở: 63

★ Kết luận với bản đồ 4:

Chi phí: DFS=BFS=Greedy=A* (vì đều tìm đường đi ra ngoài)

Số nút mở: Greedy<A*(Euclid)<DFS<A*(Manhattan)<BFS

5. Bản đồ lớn phức tạp, chật hẹp với nhiều vật cản

The height of the matrix: 15

The width of the matrix: 35

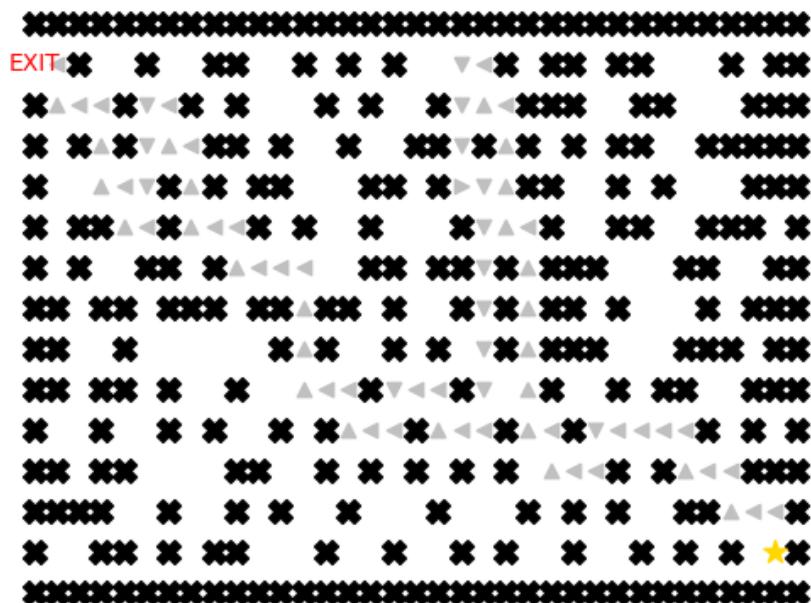
Starting point (x, y) = (13, 33) -> Ending point (x, y) = (1, 0)

a) Thuật toán tìm kiếm không có thông tin:

+ Thuật toán tìm kiếm DFS

► Chi phí: 75

► Path: [(13, 33), (12, 33), (12, 32), (12, 31), (11, 31), (11, 30), (11, 29), (10, 29), (10, 28), (10, 27), (10, 26), (10, 25), (11, 25), (11, 24), (11, 23), (10, 23), (10, 22), (9, 22), (8, 22), (7, 22), (6, 22), (5, 22), (5, 21), (4, 21), (3, 21), (2, 21), (2, 20), (1, 20), (1, 19), (2, 19), (3, 19), (4, 19), (4, 20), (5, 20), (6, 20), (7, 20), (8, 20), (9, 20), (10, 20), (10, 19), (10, 18), (9, 18), (9, 17), (9, 16), (10, 16), (10, 15), (10, 14), (9, 14), (9, 13), (9, 12), (8, 12), (7, 12), (6, 12), (6, 11), (6, 10), (6, 9), (5, 9), (5, 8), (5, 7), (4, 7), (3, 7), (3, 6), (2, 6), (2, 5), (3, 5), (4, 5), (5, 5), (5, 4), (4, 4), (4, 3), (3, 3), (2, 3), (2, 2), (2, 1), (1, 1), (1, 0)]

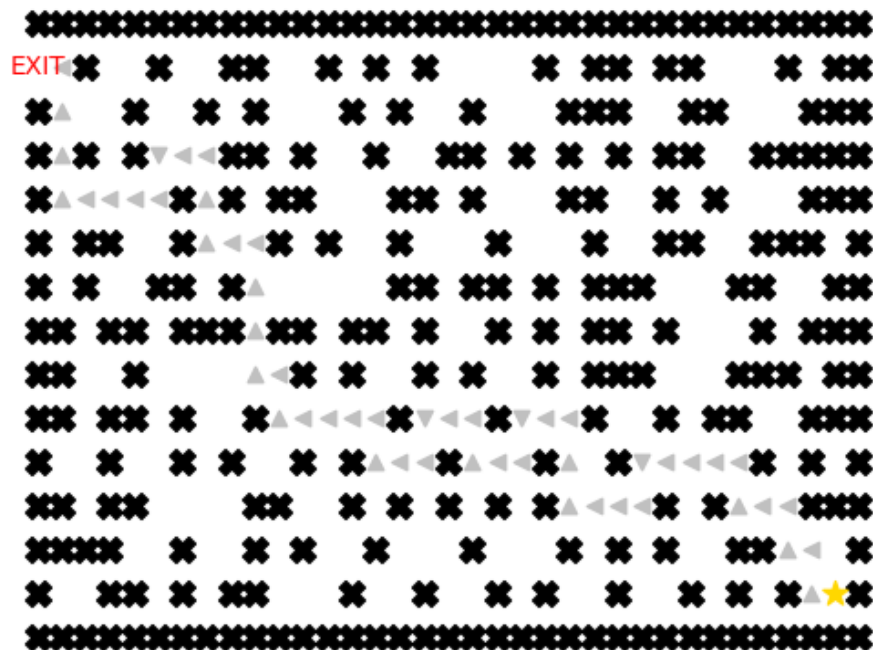


- Tìm được đường đi ra ngoài nhưng không tối ưu, số nút mở: 167

+ Thuật toán tìm kiếm BFS

- Chi phí: 53

► Path: [(13, 33), (13, 32), (12, 32), (12, 31), (11, 31), (11, 30), (11, 29), (10, 29), (10, 28), (10, 27), (10, 26), (10, 25), (11, 25), (11, 24), (11, 23), (11, 22), (10, 22), (9, 22), (9, 21), (9, 20), (10, 20), (10, 19), (10, 18), (9, 18), (9, 17), (9, 16), (10, 16), (10, 15), (10, 14), (9, 14), (9, 13), (9, 12), (9, 11), (9, 10), (8, 10), (8, 9), (7, 9), (6, 9), (5, 9), (5, 8), (5, 7), (4, 7), (3, 7), (3, 6), (3, 5), (4, 5), (4, 4), (4, 3), (4, 2), (4, 1), (3, 1), (2, 1), (1, 1), (1, 0)]



- Tìm được đường đi ra ngoài với chi phí tối ưu, số nút mở: 1278

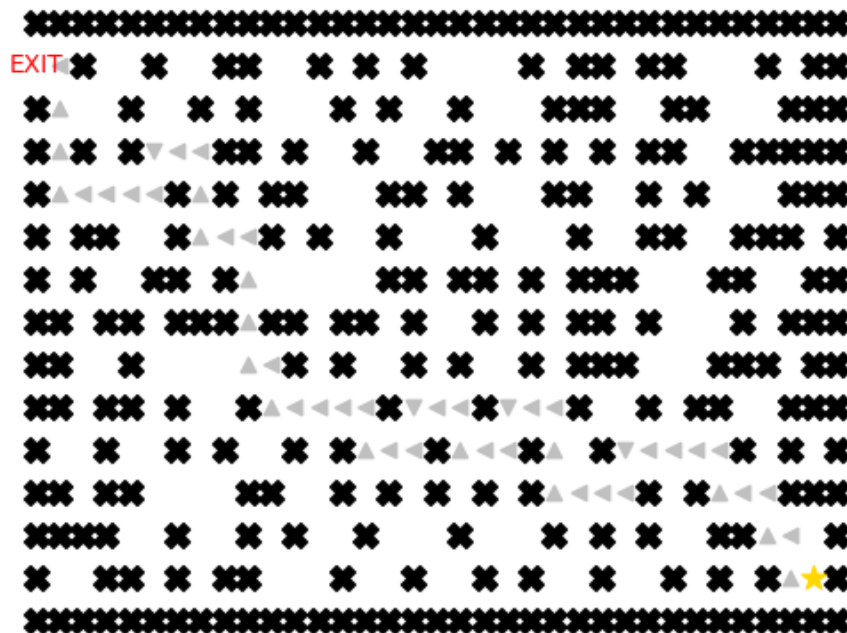
b) Thuật toán tìm kiếm có thông tin:

- Hàm heuristic đánh giá bằng khoảng cách Euclid

+ Thuật toán tìm kiếm tham lam

► Chi phí: 53

► Path: [(13, 33), (13, 32), (12, 32), (12, 31), (11, 31), (11, 30), (11, 29), (10, 29), (10, 28), (10, 27), (10, 26), (10, 25), (11, 25), (11, 24), (11, 23), (11, 22), (10, 22), (9, 22), (9, 21), (9, 20), (10, 20), (10, 19), (10, 18), (9, 18), (9, 17), (9, 16), (10, 16), (10, 15), (10, 14), (9, 14), (9, 13), (9, 12), (9, 11), (9, 10), (8, 10), (8, 9), (7, 9), (6, 9), (5, 9), (5, 8), (5, 7), (4, 7), (3, 7), (3, 6), (3, 5), (4, 5), (4, 4), (4, 3), (4, 2), (4, 1), (3, 1), (2, 1), (1, 1), (1, 0)]



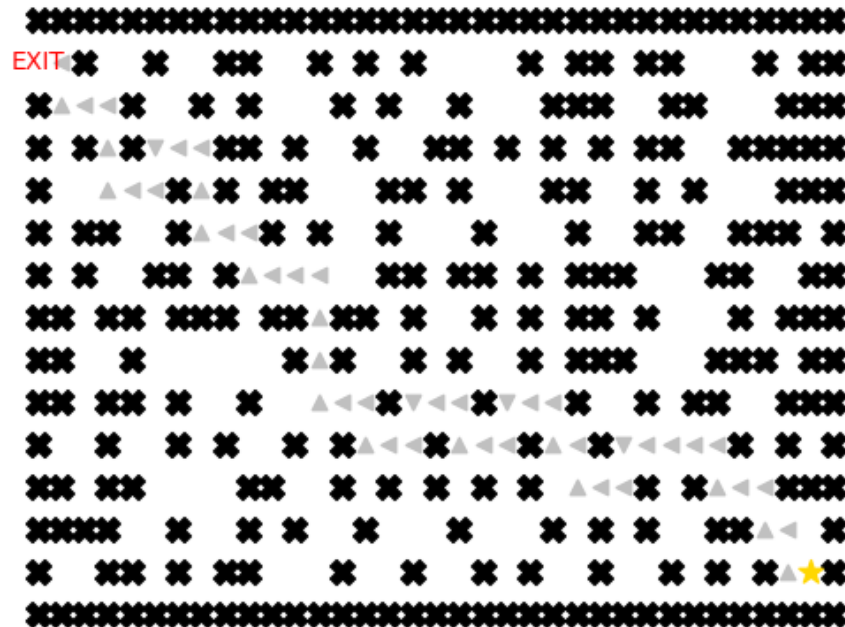
► Tìm được đường đi ra ngoài với chi phí tối ưu, số nút mở: 132

+ Thuật toán tìm kiếm A*

► Chi phí: 53

► Path: [(13, 33), (13, 32), (12, 32), (12, 31), (11, 31), (11, 30), (11, 29), (10, 29), (10, 28), (10, 27), (10, 26), (10, 25), (11, 25), (11, 24), (11, 23), (10, 23), (10, 22), (9, 22), (9, 21), (9, 20), (10, 20), (10, 19), (10, 18), (9, 18), (9, 17), (9, 16), (10, 16), (10, 15), (10, 14), (9, 14), (9, 13), (9, 12), (8, 12), (7, 12), (6, 12), (6, 11), (6, 10), (6, 9),

(5, 9), (5, 8), (5, 7), (4, 7), (3, 7), (3, 6), (3, 5), (4, 5), (4, 4), (4, 3), (3, 3), (2, 3), (2, 2), (2, 1), (1, 1), (1, 0)]



► Tìm được đường đi ra ngoài với chi phí tối ưu, số nút mở: 140

■ Hàm Heuristic đánh giá bằng khoảng cách Manhattan

+ Thuật toán tìm kiếm tham lam

► Giống trong hàm heuristic Euclid, số nút mở: 179

+ Thuật toán tìm kiếm A*

► Tìm được đường đi ra ngoài với chi phí tối ưu (cách đi khác với heuristic Euclid), số nút mở: 1278

★ **Kết luận với bản đồ 5:**

Chi phí: BFS=Greedy=A*<DFS

Số nút mở:

Greedy(Euclid)<A*(Euclid)<DFS<Greedy(Manhattan)<A*(Manhattan)=BFS

Nhận xét các thuật toán tìm đường đi trong mê cung

Thuật toán	Đặc điểm	Đánh giá chung chạy trong các bản đồ
DFS	Luôn tìm được đường đi nhưng có thể chưa ngắn nhất. Thời gian chạy lâu và tốn nhiều bộ nhớ.	Hầu như chạy không tốt trong các loại bản đồ (nếu may mắn thì sẽ gặp được trường hợp tối ưu), số lượng nút mở ra không nhiều trong 4 thuật toán.
BFS	Luôn tìm được đường đi ngắn nhất có thể. Mang tính chất vét cạn, khi không gian tìm kiếm lớn không hiệu quả vì chi phí không gian và thời gian lớn. Mang tính mù quáng, duyệt tất cả các đỉnh, không chú ý đến thông tin trong các đỉnh để duyệt hiệu quả.	Chạy tối ưu về mặt chi phí trong mọi bản đồ nhưng trả giá về mặt thời gian và bộ nhớ, số lượng nút mở ra nhiều đáng kể. Rất không hiệu quả với bản đồ trống ít vật cản.
Greedy-BFS	Luôn tìm được đường đi nhưng không chắc là ngắn nhất. Tốn nhiều thời gian nhưng tốn ít bộ nhớ nếu đường đi phức tạp. Độ tối ưu phụ thuộc vào hàm heuristic. Với hàm heuristic tốt có thể mang lại cải thiện về thời gian và không gian. Quá trình tìm kiếm có thể đi xa khỏi lời giải. Có thể bị kẹt trong vòng lặp.	Chạy tốt trong hầu hết các bản đồ đặc biệt là bản đồ trống ít vật cản đến đích. Số lượng nút mở ra trong các bản đồ hầu như ít nhất trong 4 thuật toán.

A*	Kết hợp được tính hiệu quả (nhanh) của G-BFS và tính tối ưu của UCS. Chắc chắn tìm được đường đi nếu có nhưng chưa chắc ngắn nhất. Tốn thời gian nhưng tốn ít bộ nhớ nếu đường đi phức tạp. Độ tối ưu phụ thuộc vào hàm heuristic và tính gần đúng. A* luôn mở rộng về phía đích, nhưng vẫn thực hiện đánh cược để đảm bảo sự tối ưu.	Chạy tốt trong hầu hết các bản đồ đặc biệt là bản đồ trống ít vật cản đến đích. Đa số đều tìm được đường đi ngắn nhất. Số lượng nút mở ra tương đối nhiều nếu sử dụng hàm heuristic Manhattan. Nhìn chung thuật toán A* tìm đường đi vừa hoàn chỉnh vừa tối ưu, hài hòa về các mặt trong 4 thuật toán. Thời gian chạy của thuật toán này thường thấp hơn hầu hết các thuật toán tìm kiếm khác.
-----------	---	--

II. Bản đồ có điểm thưởng

Mô tả cách giải quyết: Sử dụng thuật toán A* với chiến lược tham lam ăn hết tất cả điểm thưởng gần với vị trí hiện tại nhất (nếu điểm thưởng ở vị trí đến được) sau đó mới thoát ra ngoài.

1. Bản đồ bình thường (không phức tạp) với 2 điểm thưởng

The height of the matrix: 15

The width of the matrix: 25

Starting point $(x, y) = (10, 8)$ -> Ending point $(x, y) = (0, 11)$

Bonus point at position $(x, y) = (2, 8)$ with point -3

Bonus point at position $(x, y) = (11, 14)$ with point -10

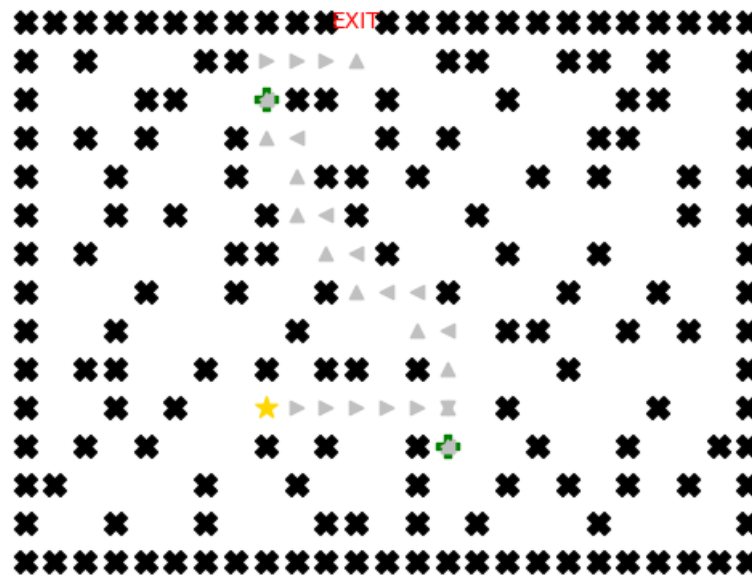
■ *Sử dụng hàm Heuristic tính khoảng cách Euclid*

► Tổng chi phí: 27

► Chi phí thật sự (đã cộng điểm thưởng): 14

► Path: [(10, 8), (10, 9), (10, 10), (10, 11), (10, 12), (10, 13), (10, 14), (11, 14), (11, 14), (10, 14), (10, 13), (10, 12), (9, 12), (8, 12), (7, 12), (7, 11), (6, 11), (6, 10), (5, 10), (5, 9), (4, 9), (3, 9), (3, 8), (2, 8), (2, 8), (1, 8), (1, 9), (1, 10), (1, 11), (0, 11)]

► Số nút mở: 222



► Hai hàm heuristic cho cách đi khác nhau nhưng đều cùng cho cùng chi phí. Số nút mở: Euclid < Manhattan

2. Bản đồ có khoảng cách từ điểm bắt đầu gần điểm kết thúc (theo đường chim bay) với 5 điểm thưởng

The height of the matrix: 15

The width of the matrix: 25

Starting point $(x, y) = (12, 19) \rightarrow$ Ending point $(x, y) = (12, 0)$

Bonus point at position $(x, y) = (13, 19)$ with point -3

Bonus point at position $(x, y) = (4, 14)$ with point -10

Bonus point at position $(x, y) = (8, 7)$ with point -5

Bonus point at position $(x, y) = (11, 7)$ with point -9

Bonus point at position $(x, y) = (8, 6)$ with point -7

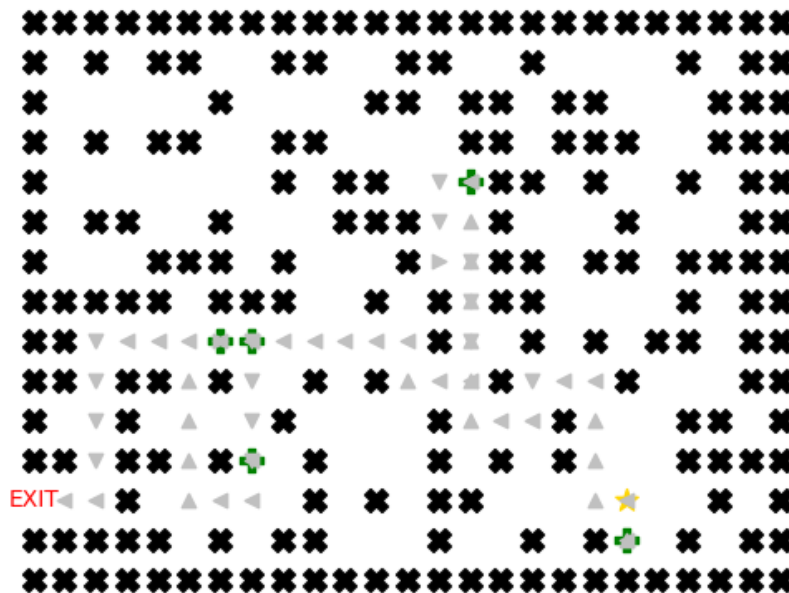
■ Sử dụng hàm Heuristic tính khoảng cách Euclid

► Tổng chi phí: 53

► Chi phí thật sự (đã cộng điểm thưởng): 19

► Path: [(12, 19), (13, 19), (13, 19), (12, 19), (12, 18), (11, 18), (10, 18), (9, 18), (9, 17), (9, 16), (10, 16), (10, 15), (10, 14), (9, 14), (8, 14), (7, 14), (6, 14), (5, 14), (4, 14), (4, 14), (4, 13), (5, 13), (6, 13), (6, 14), (7, 14), (8, 14), (9, 14), (9, 13), (9, 12), (8, 12), (8, 11), (8, 10), (8, 9), (8, 8), (8, 7), (8, 7), (8, 6), (8, 6), (8, 7), (9, 7), (10, 7), (11, 7), (11, 7), (12, 7), (12, 6), (12, 5), (11, 5), (10, 5), (9, 5), (8, 5), (8, 4), (8, 3), (8, 2), (9, 2), (10, 2), (11, 2), (12, 2), (12, 1), (12, 0)]

► Số nút mở: 172



■ Sử dụng hàm Heuristic tính khoảng cách Manhattan

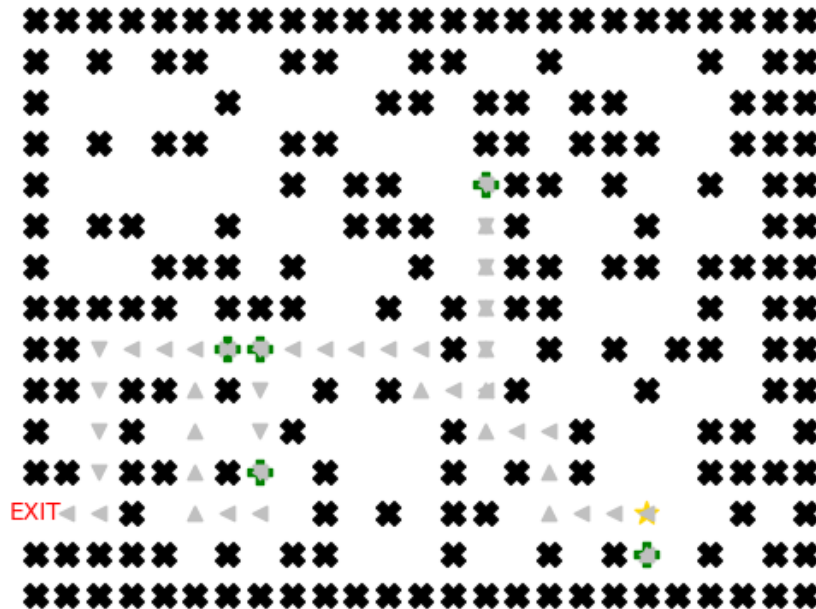
► Tổng chi phí: 49

► Chi phí thật sự (đã cộng điểm thưởng): 15

► Path: [(12, 19), (13, 19), (13, 19), (12, 19), (12, 18), (12, 17), (12, 16), (11, 16), (10, 16), (10, 15), (10, 14), (9, 14), (8, 14), (7, 14), (6, 14), (5, 14), (4, 14), (4, 14), (5, 14), (6, 14), (7, 14), (8, 14), (9, 14), (9, 13), (9, 12), (8, 12), (8, 11), (8, 10), (8, 9), (8,

8), (8, 7), (8, 7), (8, 6), (8, 6), (8, 7), (9, 7), (10, 7), (11, 7), (11, 7), (12, 7), (12, 6), (12, 5), (11, 5), (10, 5), (9, 5), (8, 5), (8, 4), (8, 3), (8, 2), (9, 2), (10, 2), (11, 2), (12, 2), (12, 1), (12, 0)]

► Số nút mở: 173



► Hàm heuristic Manhattan cho chi phí đường đi tốt hơn, số nút mở:
Euclid < Manhattan

3. Bản đồ lớn phức tạp, chật hẹp với nhiều vật cản với 10 điểm thưởng

The height of the matrix: 15

The width of the matrix: 35

Starting point (x, y) = (13, 33) -> Ending point (x, y) = (1, 0)

Bonus point at position (x, y) = (1, 18) with point -10

Bonus point at position (x, y) = (4, 3) with point -2

Bonus point at position (x, y) = (5, 21) with point -5

Bonus point at position $(x, y) = (7, 9)$ with point -1

Bonus point at position $(x, y) = (8, 28)$ with point -8

Bonus point at position $(x, y) = (10, 10)$ with point -3

Bonus point at position $(x, y) = (11, 27)$ with point -2

Bonus point at position $(x, y) = (12, 5)$ with point -12

Bonus point at position $(x, y) = (13, 10)$ with point -15

Bonus point at position $(x, y) = (13, 22)$ with point -9

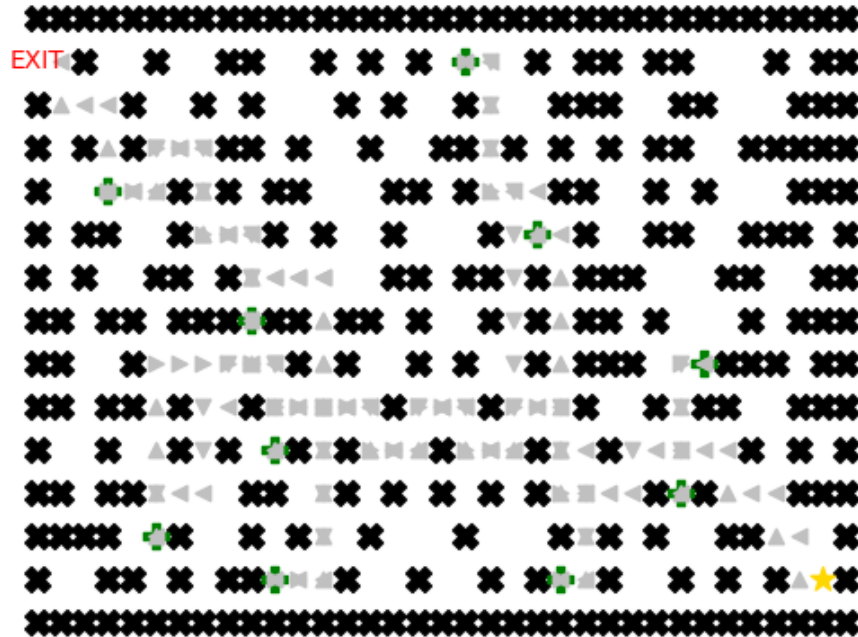
■ Sử dụng hàm Heuristic tính khoảng cách Euclid

► Tổng chi phí: 195

► Chi phí thật sự (đã cộng điểm thưởng): 128

► Path: [(13, 33), (13, 32), (12, 32), (12, 31), (11, 31), (11, 30), (11, 29), (10, 29), (10, 28), (10, 27), (11, 27), (11, 27), (10, 27), (9, 27), (8, 27), (8, 28), (8, 28), (8, 27), (9, 27), (10, 27), (10, 26), (10, 25), (11, 25), (11, 24), (11, 23), (10, 23), (10, 22), (9, 22), (8, 22), (7, 22), (6, 22), (5, 22), (5, 21), (5, 21), (4, 21), (4, 20), (4, 19), (3, 19), (2, 19), (1, 19), (1, 18), (1, 18), (1, 19), (2, 19), (3, 19), (4, 19), (4, 20), (5, 20), (6, 20), (7, 20), (8, 20), (9, 20), (10, 20), (10, 19), (10, 18), (9, 18), (9, 17), (9, 16), (10, 16), (10, 15), (10, 14), (9, 14), (9, 13), (9, 12), (8, 12), (7, 12), (6, 12), (6, 11), (6, 10), (6, 9), (7, 9), (7, 9), (8, 9), (8, 10), (9, 10), (10, 10), (10, 10), (9, 10), (9, 11), (9, 12), (10, 12), (11, 12), (12, 12), (13, 12), (13, 11), (13, 10), (13, 10), (13, 11), (13, 12), (12, 12), (11, 12), (10, 12), (9, 12), (9, 11), (9, 10), (8, 10), (8, 9), (8, 8), (9, 8), (9, 7), (10, 7), (11, 7), (11, 6), (11, 5), (12, 5), (12, 5), (11, 5), (10, 5), (9, 5), (8, 5), (8, 6), (8, 7), (8, 8), (8, 9), (7, 9), (6, 9), (5, 9), (5, 8), (5, 7), (4, 7), (3, 7), (3, 6), (3, 5), (4, 5), (4, 4), (4, 3), (4, 3), (4, 4), (4, 5), (3, 5), (3, 6), (3, 7), (4, 7), (5, 7), (5, 8), (5, 9), (6, 9), (7, 9), (8, 9), (8, 10), (9, 10), (9, 11), (9, 12), (9, 13), (9, 14), (10, 14), (10, 15), (10, 16), (9, 16), (9, 17), (9, 18), (10, 18), (10, 19), (10, 20), (9, 20), (9, 21), (9, 22), (10, 22), (11, 22), (11, 23), (12, 23), (13, 23), (13, 22), (13, 22), (13, 23), (12, 23), (11, 23), (11, 22), (10, 22), (9, 22), (9, 21), (9, 20), (10, 20), (10, 19), (10, 18), (9, 18), (9, 17), (9, 16), (10, 16), (10, 15), (10, 14), (9, 14), (9, 13), (9, 12), (8, 12), (7, 12), (6, 12), (6, 11), (6, 10), (6, 9), (5, 9), (5, 8), (5, 7), (4, 7), (3, 7), (3, 6), (3, 5), (4, 5), (4, 4), (4, 3), (3, 3), (2, 3), (2, 2), (2, 1), (1, 1), (1, 0)]

► Số nút mở: 525



■ Sử dụng hàm Heuristic tính khoảng cách Manhattan

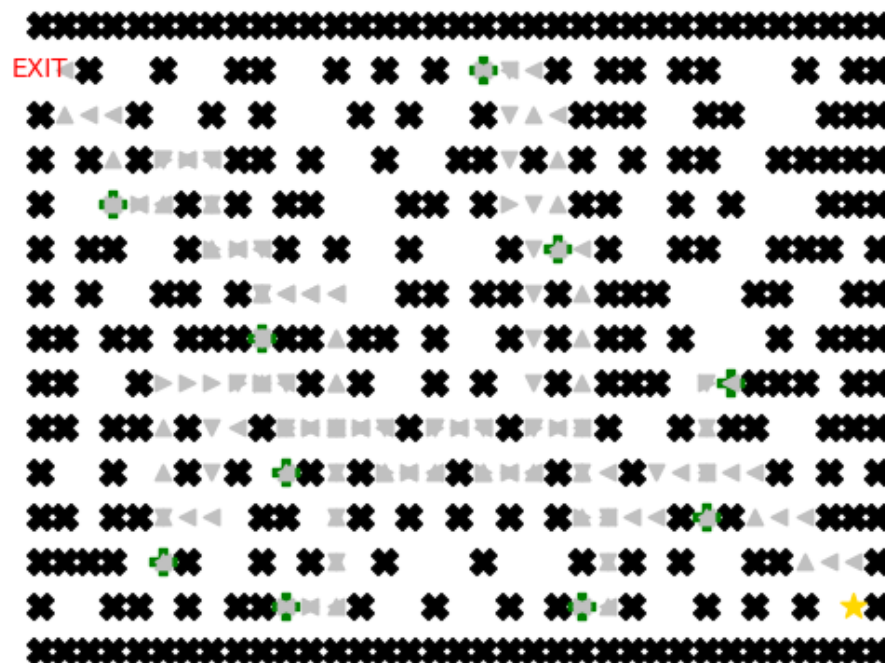
► Tổng chi phí: 195

► Chi phí thật sự (đã cộng điểm thưởng): 128

► Path: [(13, 33), (12, 33), (12, 32), (12, 31), (11, 31), (11, 30), (11, 29), (10, 29), (10, 28), (10, 27), (11, 27), (11, 27), (10, 27), (9, 27), (8, 27), (8, 28), (8, 28), (8, 27), (9, 27), (10, 27), (10, 26), (10, 25), (11, 25), (11, 24), (11, 23), (10, 23), (10, 22), (9, 22), (8, 22), (7, 22), (6, 22), (5, 22), (5, 21), (5, 21), (4, 21), (3, 21), (2, 21), (2, 20), (1, 20), (1, 19), (1, 18), (1, 18), (1, 19), (2, 19), (3, 19), (4, 19), (4, 20), (5, 20), (6, 20), (7, 20), (8, 20), (9, 20), (10, 20), (10, 19), (10, 18), (9, 18), (9, 17), (9, 16), (10, 16), (10, 15), (10, 14), (9, 14), (9, 13), (9, 12), (8, 12), (7, 12), (6, 12), (6, 11), (6, 10), (6, 9), (7, 9), (7, 9), (8, 9), (8, 10), (9, 10), (10, 10), (10, 10), (9, 10), (9, 11), (9, 12), (10, 12), (11, 12), (12, 12), (13, 12), (13, 11), (13, 10), (13, 10), (13, 11), (13, 12), (12, 12), (11, 12), (10, 12), (9, 12), (9, 11), (9, 10), (8, 10), (8, 9), (8, 8), (9, 8), (9, 7), (10, 7), (11, 7), (11, 6), (11, 5), (12, 5), (12, 5), (11, 5), (10, 5), (9, 5), (8, 5), (8, 6), (8,

7), (8, 8), (8, 9), (7, 9), (6, 9), (5, 9), (5, 8), (5, 7), (4, 7), (3, 7), (3, 6), (3, 5), (4, 5), (4, 4), (4, 3), (4, 3), (4, 4), (4, 5), (3, 5), (3, 6), (3, 7), (4, 7), (5, 7), (5, 8), (5, 9), (6, 9), (7, 9), (8, 9), (8, 10), (9, 10), (9, 11), (9, 12), (9, 13), (9, 14), (10, 14), (10, 15), (10, 16), (9, 16), (9, 17), (9, 18), (10, 18), (10, 19), (10, 20), (9, 20), (9, 21), (9, 22), (10, 22), (11, 22), (11, 23), (12, 23), (13, 23), (13, 22), (13, 22), (13, 23), (12, 23), (11, 23), (11, 22), (10, 22), (9, 22), (9, 21), (9, 20), (10, 20), (10, 19), (10, 18), (9, 18), (9, 17), (9, 16), (10, 16), (10, 15), (10, 14), (9, 14), (9, 13), (9, 12), (8, 12), (7, 12), (6, 12), (6, 11), (6, 10), (6, 9), (5, 9), (5, 8), (5, 7), (4, 7), (3, 7), (3, 6), (3, 5), (4, 5), (4, 4), (4, 3), (3, 3), (2, 3), (2, 2), (2, 1), (1, 1), (1, 0)]

► Số nút mở: 840



► Hai hàm heuristic cho cách đi khác nhau nhưng đều cùng cho cùng chi phí. Số nút mở: Euclid<Manhattan

➡ Nhận xét:

Với việc sử dụng thuật toán A* và chiến lược tham lam trên để tìm đường đi với chi phí nhỏ nhất có thể ta chỉ có thể tìm được đường đi nhưng chưa chắc tối ưu nhất. Với chiến lược heuristic này thì độ tối ưu không những phụ thuộc vào hàm heuristic mà còn phụ thuộc thêm vào vị trí của điểm thưởng và giá trị của điểm thưởng. Nếu không may mắn sẽ phải trả một chi phí rất lớn. Hàm đánh giá cần thỏa mãn 2 tính chất quan trọng: Hợp lý và Nhất quán.

* Tài liệu tham khảo:

Slide bài giảng của các thầy trong môn học

https://vi.wikipedia.org/wiki/Gi%E1%BA%A3i_thu%E1%BA%ADt_t%C3%ACm_ki%E1%BA%BFm_A*

<https://www.javatpoint.com/ai-informed-search-algorithms>

<https://labs.flinters.vn/algorithm/algorithm-cac-thuat-toan-tim-kiem-trong-ai/>

HẾT