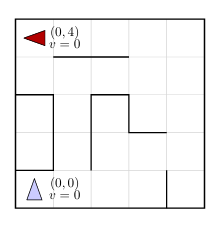
Họ tên: Nguyễn Văn Anh Tú

MSSV: 3122410445

Q1. Search and Heuristics



(a) [3 pts] Suppose the agent wants to take the leftmost path (i.e., the one that passes through (1,2)) from the start (0,0) facing north to the goal (0,4) facing west. Write down the shortest sequence of actions for it to take.

Actions: Đi từ (0,0) đến (0,4) và phải đi qua (1,2)

1: (0,0) turn right (thỏa v=0) -> quay về hướng đông

2: (0,0) faster v’ = v + 1 = 1 -> (0,1)

3: (0,1) slower v’ = v-1 = 0 -> (0,1)

4: (0,1) turn left (thỏa v=0) -> quay về hướng bắc

5: (0,1) faster v’ = v + 1 =1 -> (1,1)

6: (1,1) maintain v’ = 1 -> (1,2)

7: (1,2) maintain v’ = 1 -> (1,3)

8: (1,3) slower v’ =0 -> (1,3)

9: (1,3) turn left ( thỏa v=0) -> quay về hướng tây

10: (1,3) faster v’ = v +1 = 1 -> (0,3)

11: (0,3) slower v’ = v-1 = 0 -> (0,3)

12: (0,3) turn right (thỏa v=0) -> quay về hướng bắc

13: (0,3) faster v’ =1 -> (0,4)

14: (0,4) slower v’=0

15: (0,4) turn left (thỏa v=0) => đã đến đích hướng tây

(b) [3 pts] If the grid is M by N and the maximum speed is Vmax, what is the size of the state space? You should assume that all configurations are reachable from the start state.

State Space = 4×M×N×(Vmax​+1)

(c) [3 pts] A “child” of a state s is any other state s’ reachable via a legal action from state s. Is it possible that a state in the state space has no children? If so, give an example of such a state. If not, briefly explain why every state must have at least one child.

Có, tồn tại trạng thái mà không có trạng thái con

Giải thích: ví dụ, khi mà agent đối diện với tường và v = Vmax tức v>1 thì mọi hành động đều bất khả thi. Suy ra lúc đó sẽ không sinh ra trạng thái con

(d) What is the maximum branching factor of this problem? Draw an example state (x, y, orientation, velocity) that has this branching factor, and list the set of available actions. For example, in the above picture, if the agent was in (0, 0) facing North with a velocity of v = 0, the branching factor would be 2. The agent could turn left or right (but not go faster since it would hit a wall). Illegal actions are simply not returned by the problem model and therefore not counted in the branching factor. You do not necessarily have to use the example grid above. If you need to include a drawing of your own, label properly and make sure it fits in the solution box.

Số phân nhánh tối đa của bài toán sẽ là 3

Ví dụ: Xét với điều kiện không có tường chặn

Tại 1 ô bất kì, v = 0 và hướng không xoay vào tường: thì sẽ có 3 nhánh là turn left, right, faster

Tại v =1: sẽ là faster, slower, maintain.

(e) [4 pts] Is the Manhattan distance from the agent’s location to the exit’s location admissible? If not, draw an example state (x, y, orientation, velocity) where this heuristic overestimates at that state, and specify: 1) the heuristic value at that state and 2) the actual cost from that state to the goal. You do not necessarily have to use the example grid above. Make sure to label your drawing, including the goal state (location, orientation, speed) and action sequence, and fit it into the solution box.

**Không — Manhattan distance không nhất thiết admissible.**  
Lý do: Manhattan đo số ô (số bước ô) còn chi phí ở bài này là số ***hành động*** (actions); một hành động có thể di chuyển nhiều ô (v' > 1). Vì vậy có những trạng thái mà ta có thể vượt nhiều ô trong một hành động, tức là **số hành động thực tế cần** để đến đích có thể ít hơn giá trị Manhattan — nghĩa là Manhattan có thể **over-estimate** chi phí thực, nên không đảm bảo tính admissible.

(f) [4 pts] Is the following heuristic admissible? Manhattan distance / Vmax.

If yes, state why. If not, draw an example state (x, y, orientation, velocity) where this heuristic overestimates at that state, and specify: 1) the heuristic value at that state and 2) the actual cost from that state to the goal. You do not necessarily have to use the example grid above. Make sure to label your drawing, including the goal state (location, orientation, speed) and action sequence, and fit it into the solution box.

Có heuristic là có thể chấp nhận

Độ đo manhattan D biểu thị số ô cần đi đến đích nếu chỉ tính khoảng cách. Trong bài toán này, mỗi hành động chỉ có thể giúp agent di chuyển tối đa Vmax ô (vì tốc độ lớn nhất là Vmax)

Vì vậy, để đi được quãng đường D ô, agent ít nhất phải cần D/Vmax hành động.

Chi phí thật sự (số hành động) có thể nhiều hơn vì agent còn phải tăng tốc, giảm tốc hoặc rẽ, nhưng **không bao giờ ít hơn** D/Vmax​.  
Do đó, heuristic này **không bao giờ đánh giá cao hơn chi phí thật**, nên **admissible.**

(g) [1 pt] If we used an inadmissible heuristic in A\* Tree search, could it change the completeness of the search? Assume the graph is finite and the heuristic is non-negative.

NO

(h) [1 pt] If we used an inadmissible heuristic in A\* Tree search, could it change the optimality of the search? Assume the graph is finite and the heuristic is non-negative.

Yes

(i) [3 pts] Which of the following may be a good reason to use an inadmissible heuristic over an admissible one? Select all that apply.

 **An inadmissible heuristic may be easier to compute, leading to a faster state heuristic computation time. — Đúng.** Heuristic đơn giản, nhanh tính có thể tăng tốc tổng thời gian tìm kiếm.

 **An inadmissible heuristic can be a closer estimate to the actual cost (even if it’s an overestimate) than an admissible heuristic, thus exploring fewer nodes. — Đúng.** Một heuristic “tham” hơn đôi khi dẫn tới ít nút được mở rộng hơn, giảm thời gian/nhớ thực tế, dù không bảo đảm tối ưu.

Q2. Search Nodes

a) Breath-first search

BFS duyệt theo độ sâu (depth). Nó sẽ tìm thấy nút đích nông nhất . Nó sẽ duyệt tất cả các nút n có độ sâu nhỏ hơn . **Bất đẳng thức:** Tất cả , sao cho:

b) Uniform Cost Search

UCS duyệt theo chi phí g(n). Nó sẽ tìm thấy nút đích có chi phí thấp nhất (chi phí ). Nó sẽ duyệt tất cả các nút n có chi phí g(n) nhỏ hơn . **Bất đẳng thức:** Tất cả n, sao cho:

(c) A Search (h là consistent*)* duyệt theo . Nó sẽ tìm thấy nút đích (chi phí ). Khi là consistent (và do đó là admissible), được đảm bảo duyệt tất cả các nút n có . **Bất đẳng thức:** Tất cả n, sao cho:

Q3. Search a Graph

a)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **A-C-E-G** | **A-C-E-F-G** | **A-B-D-E-F-G** | **Other** |
| UCS | (i) | (ii) | (iii) **X** | (iv) **X**¹ |
| Greedy với h1 | (v) | (vi) | (vii) **X** | (viii) **X**¹ |
| Greedy với h2 | (ix) **X** | (x) | (xi) | (xii) |
| A\* với h1 | (xiii) | (xiv) | (xv) **X** | (xvi) **X**¹ |
| A\* với h2 | (xvii) | (xviii) | (xix) **X** | (xx) |
| ¹ *Ghi chú: Đường đi A-B-D-E-G (chi phí 11) cũng có thể được trả về bởi UCS và A* *với h\_1 tùy thuộc vào việc xử lý thứ tự ưu tiên (tie-breaking). Greedy với h\_1 cũng có thể trả về đường này nếu E được ưu tiên hơn D (vì cùng h\_1=3).* |  |  |  |  |

b) Chi phí của đường đi tối ưu cho UCS từ A đến G là 11

c) có admissible ?

 **Admissible:** Không. Chi phí tối ưu từ A đến G là 11. (A) = 12. Vì , nó không admissible.

 **Consistent:** Không. Một heuristic không admissible thì không thể consistent. (Kiểm tra: cạnh A->B, chi phí 2. là Sai).

d) Admissible -> yes, Consistent -> no

Q4. Search: multiple Choice and Short Answer Questions

a)

False

True

True

False

True

False

b)

|  |  |
| --- | --- |
| **State** | **h(s)** |
| S | **3** |
| A | **1** |
| C | **2** |
| G | 0 |

Q5. Alpha-Beta Pruning with Iterative Deepening

a) D then C then B

b) Nodes F,G,I,J

C) No