CSSE 413: Planning and Search Programming Assignment

Name: \_\_\_\_\_\_\_\_\_\_Doris Chen\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please answer the following questions and submit a pdf copy of this document with your code.

1. [35 pts] Fill in the following table using BFS/BFSM:

|  |  |  |
| --- | --- | --- |
| Map | Length of solution | Number of states placed on open |
| 1 | 9 | 10 |
| 2 | N/A | 80 |
| 3 | 17 | 83 |
| 3, with cycle detection only, i.e. only checks “open” for duplicate entries. | 17 | 400 |
| 3, with duplicate detection only, i.e. only checks “closed” for duplicate entries. | 17 | 22217 |
| 4 | 44 | 45 |
| 5 | N/A | 43 |
| 6 | 88 | 2025 |
| M1 | 11 | 35 |
| M2 | 32 | 68 |
| M3 | 106 | 390 |
| M4 | 31 | 94 |
| M5 | 68 | 159 |

1. [3 pts] For map 1, explain the number of states that are placed on open. Why do we have as many as we get?

Because there are only 10 valid grids that are reachable. In my code, the robot cannot go back to the visited grid.

1. [3 pts] For map 2, explain the number of states that are placed on open. Why do we have as many as we get?

Map 2 is a 8\*10 grid, so 80 possible states.

1. [4 pts] On map 6, does your robot hug the border of the world? If so, explain why it does not simply go across the world. Alternatively, if your robot goes across the world, why is that so?

Yes, it hugs the border. I think it will vary when the order of checking the direction, I checked in the order of going up, right, down, and left. The robot will go to the first valid grid.

1. [15 pts] Fill in the following table using astar/astarM:

|  |  |  |
| --- | --- | --- |
| Map | Length of solution | Number of states placed on open |
| 1 | 9 | 10 |
| 2 | N/A | 80 |
| 3 | 17 | 51 |
| 4 | 44 | 45 |
| 5 | N/A | 43 |
| 6 | 88 | 261 |
| M1 | 11 | 27 |
| M2 | 32 | 54 |
| M3 | 106 | 271 |
| M4 | 27 | 55 |
| M5 | 68 | 133 |

Notes on scoring the table above. The 15 points are for the non-greyed out rows.

For MapM3, the points will be distributed as follows. If the heuristic is not admissible, 0 points.

* 1. Any non-trivial heuristic: 5 pts
  2. <= 1600 nodes expanded: 10 pts
  3. <= 1200 nodes expanded: 15 pts
  4. <= 800 nodes expanded: 20 pts

1. Please explain the heuristic you used for problem MapM3.

I used Manhattan distance which is the sum of the absolute differences between the two vectors,

For MapM5, the points will be distributed as follows. If the heuristic is not admissible, 0 points.

1. <= 15000 nodes expanded: 5 pts
2. <= 12000 nodes expanded: 10 pts
3. <= 9000 nodes expanded: 15 pts
4. <= 7000 nodes expanded: 20 pts
5. Please explain the heuristic you used for problem MapM5.

In order to find the closest target first, I will calculate it first to get the closest target and recalculate Manhattan distance each time when the robot reaches to different node. Thus, my robot will reach targets on the first two rows and go down to the third row to reach the targets at the very bottom row.

1. [Extra credit, up to 10 points] For MapM6 anything goes. See whether you can muscle your way through this problem to finding a solution, any solution, especially sub-optimal ones. Please explain how you solved this problem.