CSSE 413: Planning and Search Programming Assignment

Name: \_\_\_Joseph Peters\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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 | 9 |
| 2 | na | 80 |
| 3 | 17 | 82 |
| 3, with cycle detection only, i.e. only checks “open” for duplicate entries. |  |  |
| 3, with duplicate detection only, i.e. only checks “closed” for duplicate entries. |  |  |
| 4 | 44 | 44 |
| 5 | na | 43 |
| 6 | 88 | 2024 |
| M1 | 28 | 12 |
| M2 | 60 | 32 |
| M3 | 365 | 106 |
| M4 | 55 | 27 |
| M5 | 127 | 68 |
| M6 | 295 | 192 |

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?

There are nine states, not including the target, that the robot can get to and they are all in a straight line, thus we check every and only those 9 states. My program terminates without adding the target to open.

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?

There are 80 states that the robot can get to which don’t contain a path to the goal so it checks every state.

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?

My robot checks up, down, left, right, in that order. Thus the down path gets to the other side of the board first and is led back up a track in the reverse order so that it’s path runs down the left side and then runs along the bottom to the goal.

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

|  |  |  |
| --- | --- | --- |
| Map | Length of solution | Number of states placed on open |
| 1 | 9 | 9 |
| 2 | na | 80 |
| 3 | 17 | 33 |
| 4 | 44 | 44 |
| 5 | na | 43 |
| 6 | 88 | 175 |
| M1 | 11 | 27 |
| M2 | 32 | 52 |
| M3 | 111 | 169 |
| M4 | 32 | 60 |
| M5 | 91 | 140 |

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 find the closest target to the robot in terms of direct distance and then move to whichever spot is closest to that target until the target is reached. Repeat until all targets are touched.

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.

Same as above.

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.

Same as above. Uses 228 steps and looks at 543 states.