MP 1: Search

ABHI KAMBOJ

ECE 448

Section R: Prof. Hockenmaier

2/3/2019

Section I:

A state is a description of the environment around the agent and the actions it can take. A node is the representation of the state in a search tree where actions correspond to branches in the tree leading to other states or nodes. In my case, the states were represented as possible x,y locations the agent could be on in the 2D plane of the maze and the possible actions the agent could take so that it doesn't run into the wall. The nodes were ordered pairs (x,y), or in python 2-tuples of the location. A dictionary was used to keep track of which tuples the agent has visited so there was no repetition, and a list was used to store the current set unexpanded nodes or the frontier. The dictionary held pairs with the key being a node and the value being the previous node that caused that node to be added to the frontier. Then when the goal is reached, the dictionary can be traversed from the goal to the start state and a path can be created from the set of visited nodes.

The frontier was a list; however, it was implementing a different data structure for each algorithm. For Breadth First Search (BFS), the list was used as a First In First Out (FIFO) queue frontier holding all the nodes that needed to be visited. Depth first search was similar except used a Last In First Out (FIFO) stack as the frontier. The greedy algorithm traversed the frontier every step to find the node with the smallest Manhattan distance and expand it. Finally, the astar algorithm also had the frontier store how far each node is from the start and traversed the frontier to find the smallest Manhattan distance in cost from goal plus cost to goal.

Section II:

For part 1, I implemented the A* and Greedy BFS algorithms both with Manhattan distance heuristic. This assumes that the cost to goal is just the sum of the absolute value of the difference between the goal's x coordinate and the agents x coordinate and the goal's y coordinate and the agent's y coordinate: $h_{manhattan} = |x_{goal} - x_{agent}| + |y_{goal} - y_{agent}|$. For part 2, I implemented the A* algorithm with the Euclidean distance heuristic which assumes the cost to goal is the difference in distance between the goal and agent's position in 2D space: $h_{euclidean} = sqrt((x_{goal} - x_{agent})^2+(y_{goal} - y_{agent})^2)$.

Both heuristics are admissible. An admissible heuristic must never overestimate the cost to goal, so the heuristic's assumed cost to goal must be less than or equal to the actual distance. This is true for the Manhattan distance because if there were no walls and the agent can only move up, right, down and left the quickest way to the goal would be finding the difference in x and y locations and adding them which is what Manhattan distance does. The Euclidean distance heuristic can be proven to be admissible by the triangle inequality. No one side of a triangle can be greater than its other two sides and the agent cannot move diagonally. Therefore, the Euclidean distance can be seen as the hypotenuse of a triangle or sum of smaller triangles in which the robot would have to traverse those triangles' legs to reach the goal.

For the multiple dots section (Part 2), the Euclidean distance heuristic applied to all the goal states. The heuristic h at a given node n was the minimum of the distances from that node to all the other goals. Since the Euclidean distance to a single goal is an admissible heuristic, the minimum distance to one of multiple goals must also be admissible because it is the minimum of underestimates which still must be an underestimate compared to the actual distance. As objectives were visited, they were removed from the list of objectives that was being used in the heuristic until there were no objectives left, in which case the shortest path was found.

Section III:

BFS

Medium Maze

Results

Path Length: 111

States Explored: 645

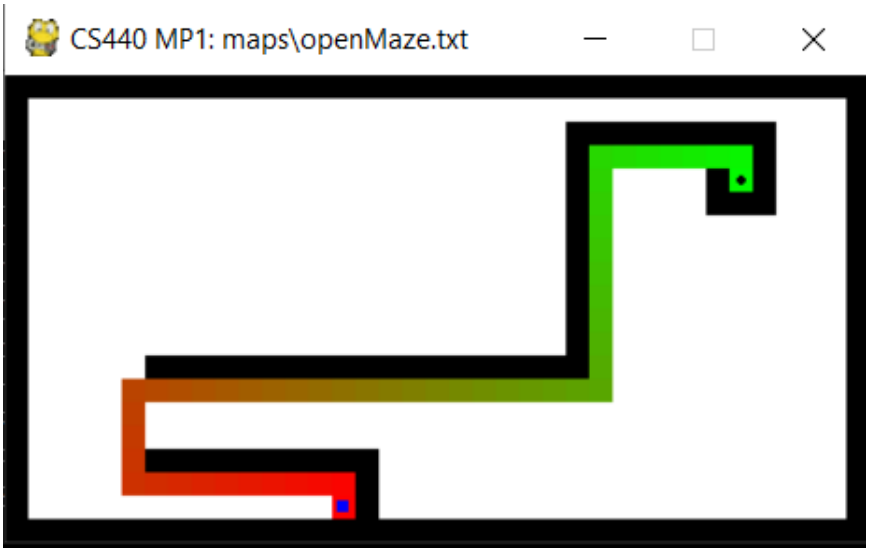
Big Maze

Results

Path Length: 183

States Explored: 1285

Open Maze



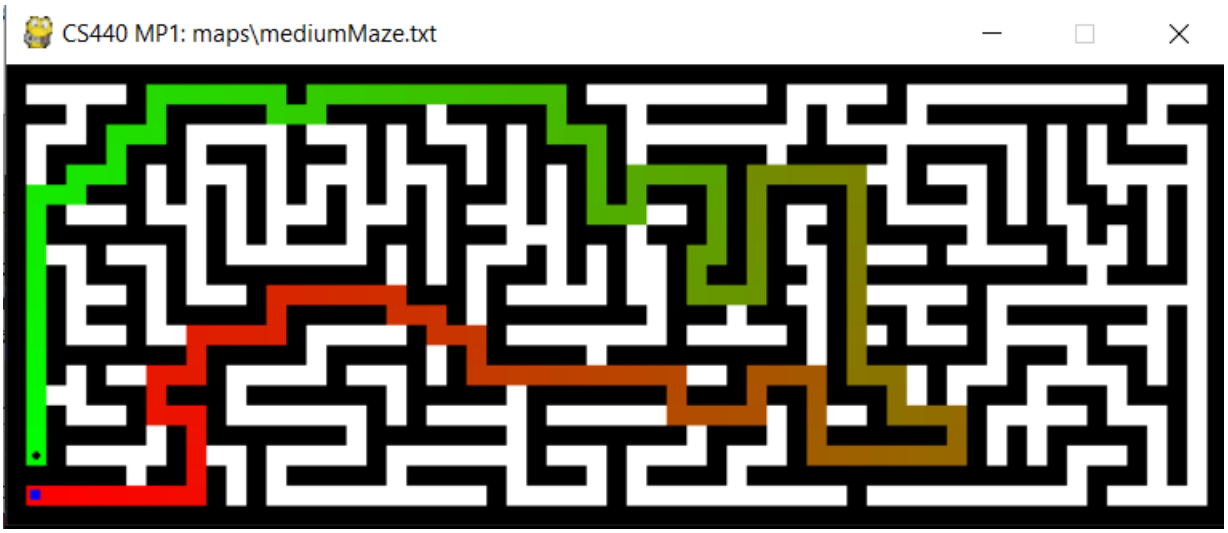
Results

Path Length: 52

States Explored: 558

DFS

Medium Maze



Results

Path Length: 175

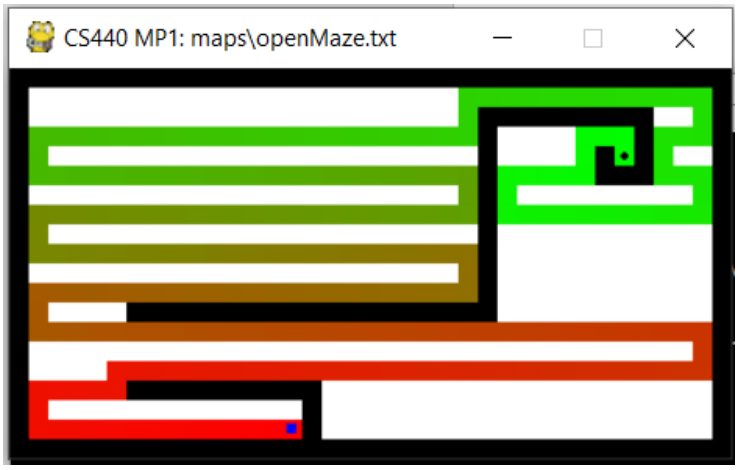
States Explored: 372

Big Maze

Results

Path Length: 537

States Explored: 985



Open Maze

Results

Path Length: 252

States Explored: 277

Greedy Algorithm

Medium Maze

Results

Path Length: 147

States Explored: 347

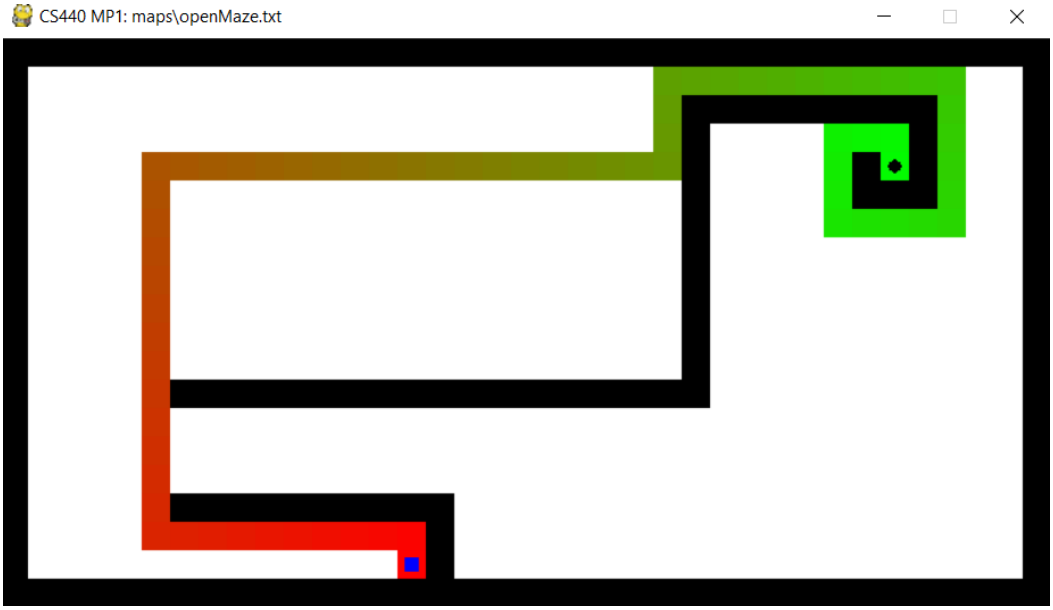
Big Maze

Results

Path Length: 277

States Explored: 456

Open Maze



Results

Path Length: 70

States Explored: 95

Astar Algorithm

Medium Maze

Results

Path Length: 112

States Explored: 417

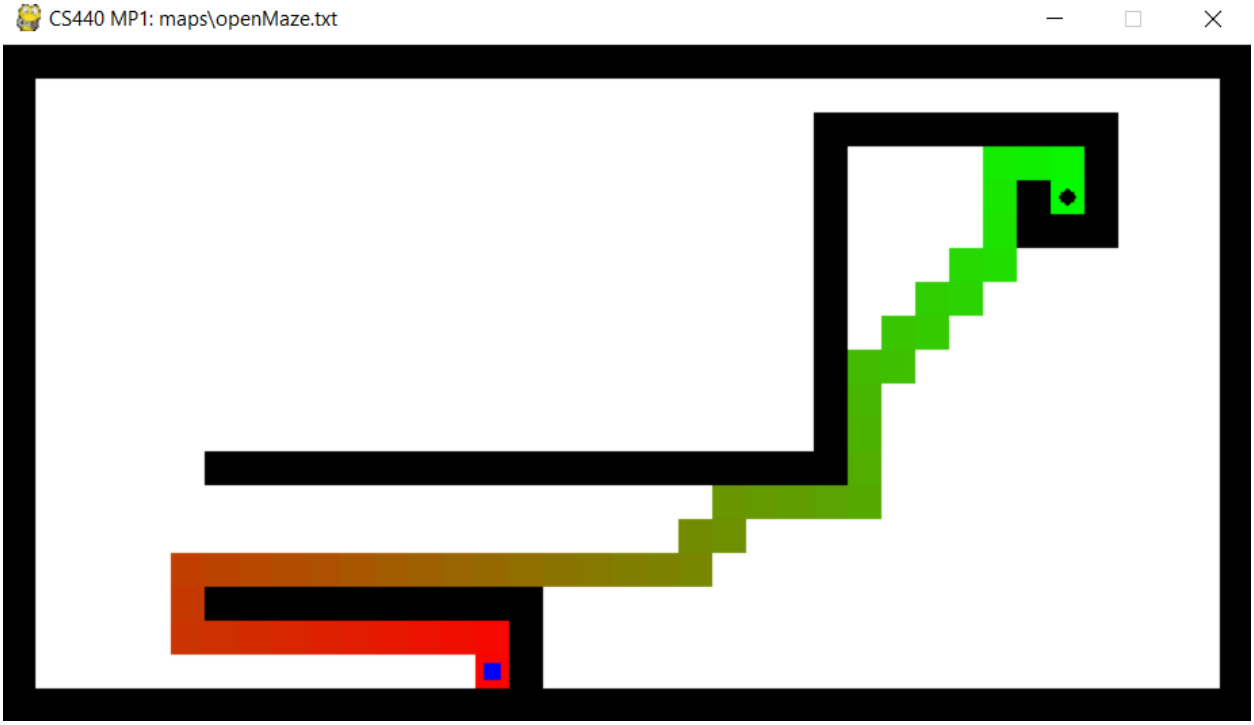
Big maze

Results

Path Length: 184

States Explored: 1265

Open Maze



Results

Path Length: 53

States Explored: 436

Section IV:

Astar

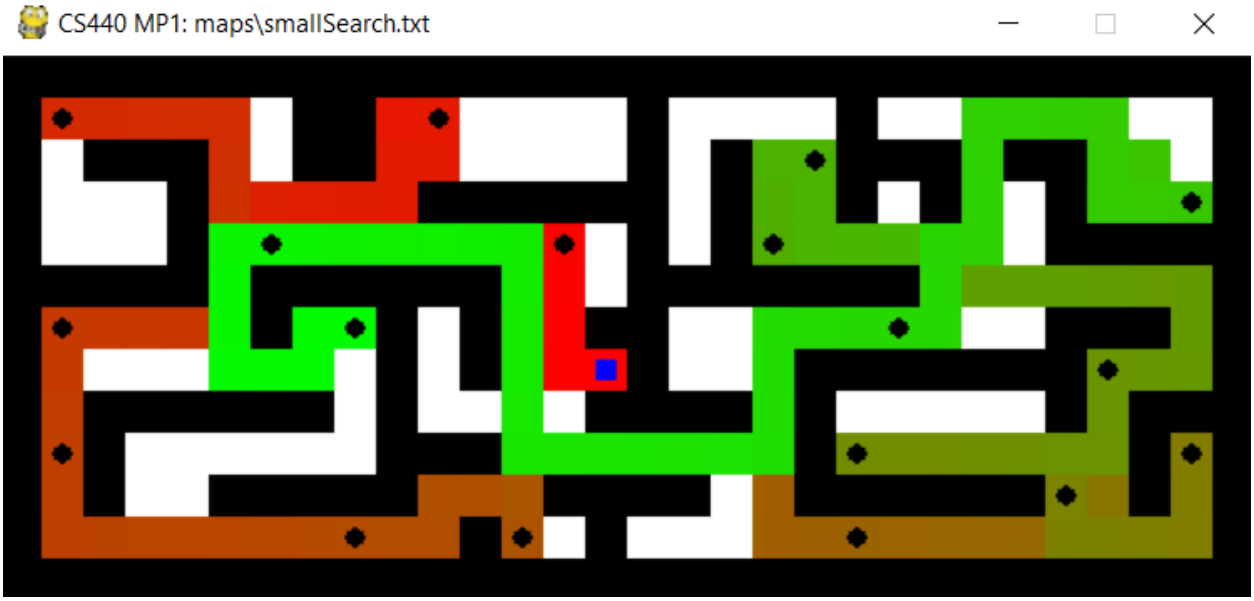
Tiny

Results

Path Length: 48

States Explored: 33

Small

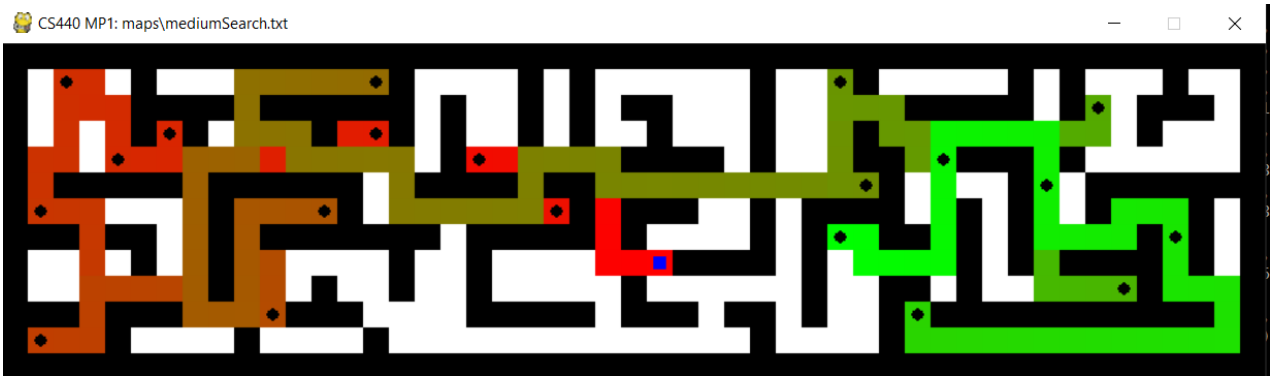


Results

Path Length: 213

States Explored: 196

Medium



Results

Path Length: 272

States Explored: 276

EXTRA CREDIT/CONTRIBUTION:

I was not able to find a team in time, which hopefully will not put me at a disadvantage. I, Abhi Kamboj, have complete all of the assignment myself.