Lab 2 - Uninformed Search

Aidan Schneider

CS 2400 031

03/21/23

1. The performance of both greedy and A* are roughly the same however A* used about 1000 more units of memory and took about 0.001 seconds longer. Even though A* took up a little more in resources, A* got a path length of 11 compared to Greedy's path length of 15. The reason Greedy is longer is due to this maze having two different directions to go and the longer path has nodes closer to the goal than the shorter path, so Greedy prefers that path and suffers because of it.
2. When it comes to performance, DFS does worse than A* and Greedy half the time on mazes 3-6. The mazes that DFS does worse on are 3 and 5. In these cases DFS takes 3x more memory on maze 3 and 5x more memory on maze 5 than both A* and Greedy. Timewise DFS takes about 2.6x longer on maze 3 and 5x longer on maze 5. On mazes 4 and 6 however DFS does roughly the same performance as both A* and Greedy. The reason for the worst case performance difference is probably due to DFS not start immediately going down the correct path in its search and instead has to go through every node in the maze to find the goal. However, since A* and Greedy use the priority queues to go to nodes that are closest to the goal, these search methods can get to the goal state quicker. In the DFS best case it is more than likely that the searches are going down the correct path right away and finds the goal just as quick as A* and Greedy because they all search the same path at the same speed.
3. For BFS, it does worse than A* and Greedy for every single maze. This is most likely due to BFS having to search every single level of the search tree causing the BFS approach to search a lot of nodes that are useless to the search. A* and Greedy don't have this problem because of their priority queues only putting the important nodes in

the front of the queues allowing to ignore any nodes that aren’t helpful in the search thus shortening computation time and memory.