Nathaniel Coleman Assignment 09 Analysis CS2420

When you are satisfied that your program is correct, write a brief analysis document. The analysis document is 10% of your assignment 9 grade. Ensure that your analysis document addresses the following.

1. Who is your programming partner? Which of you submitted the source code of your program?

2. Evaluate your programming partner.

3. Does the straight-line distance (the absolute distance, ignoring any walls) from the start point to the goal point affect the running time of your algorithm?

No, it cannot be said that there is a correlation or causation between the straight-line path and runtime of our algorithm. Although having a longer straight-path distance will guarantee a relatively longer runtime, a very short straight-line path cannot guarantee a short run time.

4. Explain the difference between the straight-line distance and the actual solution path length. Give an example of a situation in which they differ greatly. How do each of them affect the running time of your algorithm? Which one is a more accurate indicator of run-time?

The straight-line distance depends only on the locations of the start and goal node and not on the complexity of the maze. The solution path length is vice versa. Suppose there is a large and complex maze where the goal and start nodes are separated only by a wall, but the shortest path traverses the perimeter of the maze. The straight-line distance is short, but the runtime of my algorithm could potentially be very long. A better indicator of runtime will be the solution path distance.

5. Assuming that the input maze is square (height and width are the same), consider the problem size, N to be the length of one side of the maze. What is the worst-case performance of your algorithm in Big-Oh notation? Your analysis should take in to account the density of the maze (how many wall segments there are in the field). For example, a completely open field with no walls other than the perimeter is not dense at all, as opposed to the example maze given "bigMaze.txt", which is very dense. There is no one correct answer to this since solutions may vary, but you must provide an analysis that shows you are thinking about the problem in a meaningful way related to your solution.

A worst-case scenario for any size maze would have the goal node be found last. Twisting and turning within the maze would have no effect, nor would diverging paths. Since BFS can only check one path by one node in one iteration, it matters not whether there are many paths within its queue at one time. In conclusion, I believe the worst-case scenario would be a maze more like a labyrinth, with the shortest path

being the only path, perhaps spiraling to close in on the center. For a maze of this design with size N, approx. half the nodes would be walls and half would be traversable, leaving $N^2/2$. BFS would have to check all of these nodes exactly once, so the Big O notation for a worst-case scenario is N^2.

6. How many hours did you spend on this assignment?

7.