Brian Park

Cs 2420

Assignment 09

1. My programming partner was Nate Steadman. I am submitting the source code for our program.
2. Once again Nate and I made a great duo and were able to write code and crush bugs in a timely manner all while maintaining a good attitude.
3. The straight line distance ignoring walls from one side of a graph to another can influence the runtime. Our algorithm is not noticeably slowed by increasing the straight line distance from the start to the finish, however, the longer the distance, the more nodes must be added to the queue regardless of the density of edges. So by definition it should increase runtime as more calculations are done. Although other factors such as number of edges and vertices influence runtime much more. The bigO behavior of our breadth first algorithm relies on the branching factor, which relies on the number of nodes and the number of edges each node has.
4. The actual distance and the solution length differ significantly. The solution length depends on the distance from the start to the finish ie number of nodes, as well as the number of edges that are encountered on the way to the goal that branch off of the visited nodes. These two would differ incredibly in different cases. Is cases such as a straight maze with no walls, they would be close. However, if the maze was a giant empty square with the start and goal at diagonal corners, the actual path length would be very inaccurate in terms of runtime estimation. This is because the actual solution must check all paths, branching from each edge on each node, until it finds the shortest one. So if there is a large permutation of

possible paths, the algorithm will take much longer. Thus the straight line distance would be a poor indicator of runtime.

5. The worst case scenario for an input maze of equal dimensions where each dimension is N would be represented by the area of the square if there were no walls inside of it. That is it would be N*N because every single node would have to be checked in the process of finding the goals. I would rather express the notation in relation to the number of nodes and number of edges between them. For example the BigO behavior would be relative to the number of nodes + the number of edges. This is due to the introduction of walls (a dense maze) having less locations to visit. If each node has 4 pointers but two point to walls and one is going to a visited node, there is only one node to add to the queue and this fewer nodes to check. This point is touched on in number 4 with a straight maze vs an empty one.
6. Approximately 8 hours were spent on this assignment.