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

Kevin Claiborne, he submitted the source code.

2. Evaluate your programming partner.

He was generally good at what I was bad at and less good at what I was really good at. While I quickly found a solution of the problem given, he was very good at finding ways to simplify and test the code which I am not good at.

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?

Yes, the further away the two nodes are, the longer the program will take to run since it will require more checks of neighbors and expansion of the queue.

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?

Straight line distance is the actual distance between the two points ignoring walls, actual solution path is the shortest path while not passing through any walls. An example of where they differ greatly is where the start and goal are only 2-3 spaces apart but are separated by a long wall, so the actual path has to take a long path around the wall in order to get to the goal. Both increase the run time because they both require the program to consider more neighbors and expand the queue if they are larger, so the longer either of them is the longer the running time. Actual solution path is a more accurate indicator of run-time since it will be greater than or equal to the straight-line-distance based on where the walls are placed, and the run-time is based on which path needs to consider more neighbors and queue more.

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.

I'm thinking it would be N^2 since in worse case there would be no walls except for those defining the boundaries, which in that case would require a far larger amount of path and neighbor checking before a shortest route is found because you would have to check almost all the empty nodes before reaching goal with exception of a few right before the goal. In other words, the more dense a maze is the better the performance the path finding will have.

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

5-10 hours