William Frank
11/2/16
Assignment 09 Analysis

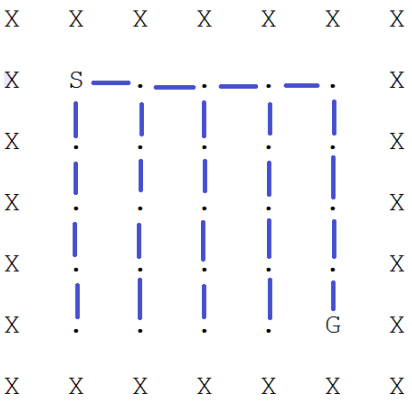
1. My programming partner was Elliot Lee, and I am submitting the source code.

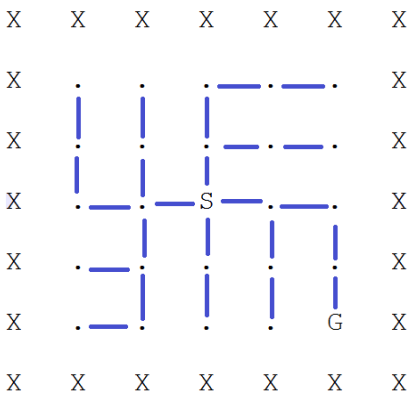
2. My programming partner and I worked together very well. We were able to complete the assignment quickly and efficiently. We helped each other complete the assignment and were able to contribute equally.

3. In general, the straight-line distance between the start and end points does not affect the run-time of our algorithm. While the two points may be very close together, the algorithm could potentially end up having to traverse the majority of the maze due to walls. Meanwhile, two points at the maximum distance could find a path in less time if there are minimal obstructions along a path with the same length as the straight-line distance.

4. The straight-line distance between two points is the shortest possible route if there are no obstructions, while the actual path may have to account for walls. The start and goal points could potentially be separated by only one wall, but still require a path that traverses the majority of the tiles in the maze. In general, the straight-line distance does not affect the run-time of the algorithm, while the actual solution path length will give you the minimum number of nodes that were checked to find the solution. For this reason the actual solution path length is a far better determiner of the algorithm's run-time.

5. If the maze is completely non-dense (no walls) then the algorithm will run in approximately O(N^2) time. This is because the algorithm will end up checking between 1 and N^2 nodes, and if this is the worst case then the most nodes will be checked. If the start and goal points have a straight-line distance of at least N-1 and there are no walls then the maximum amount of nodes will be checked, the maximum amount of paths will be created. The reason the start and goal nodes must have a straight-line distance of N-1 is to allow for the maximum number of branches to be created. If the two points are any closer then the solution will be found before every possibility can be checked. The reason it is N^2 is because the start and goal points can be arranged so that a path will check every possible tile before finding the solution. There are several configurations that will achieve this, but two are pictured below.





These imagines depict two possible worst-case scenarios for the algorithm. However, the addition of density only serves to make the algorithm more efficient. Every time a path reaches a wall that means there is one less path created, and one fewer path to check. The worst case performance of the algorithm is O(N^2), but every time a path reaches a wall that means there is one less path to update. Even if the solution path is longer, the algorithm will ultimately have to check fewer nodes, and as a result will terminate faster.

6. 5-10 hours