Kevin Claiborne

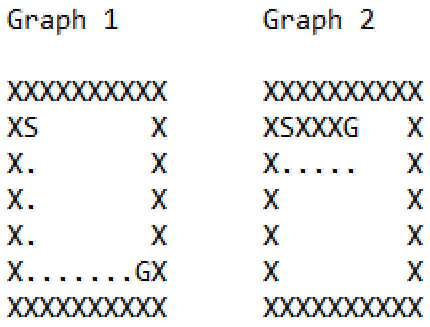
UID: u1080787

Assignment 09 Analysis

1. For this assignment, I continued to work with Daniel Zhu. I submitted the source code.

2. I continue to be impressed with Daniel ability to pick things up so quickly. Without much thought at all, he can write code and problem solve much faster than I can. He is very good and easy to work with as well, which makes for a nice partnership. I can always count on him to be responsive and easy to schedule with. When we meet up we are quick and we always trade off very evenly. I also always very appreciative of him explaining his steps and thought process with me while we are working together.

3. The straight-line distance does influence the run-time of the algorithm. The shorter the straight-line distance is, the less vertices and edges will need to be checked. Because the way breadth first search is implemented it continually spreads through the neighbors until a path is found. The closer the straight-line path is the faster a path will be found.

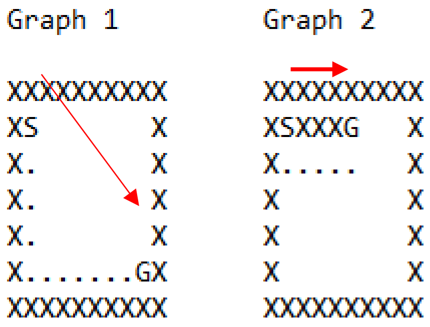


```
Graph 1         Graph 2

XXXXXXXXXX      XXXXXXXXXX
XS       X      XSXXXG   X
X.       X      X.....   X
X.       X      X        X
X.       X      X        X
X.......GX      X        X
XXXXXXXXXX      XXXXXXXXXX
```

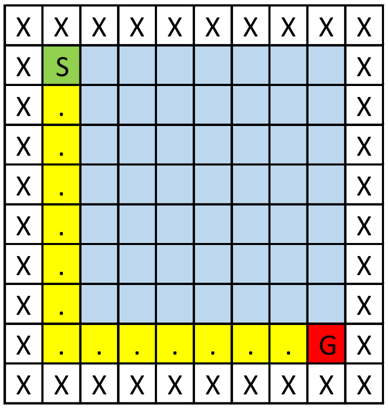
As you in see in the graphs above, Graph 2 will take less time to search, as less neighbors are checked overall.

4. The difference between the straight-line distance and the actual distance that needs to be traveled is that the actual distance traveled will generally be longer than the straight-line distance. Hence going from point S to point G on the following two graphs, theoretically, if you could draw a line directly from S to G it would be shorter than the path we can take there.



```
Graph 1         Graph 2

XXXXXXXXXX      XXXXXXXXXX
XS       X      XSXXXG   X
X.       X      X.....   X
X.       X      X        X
X.       X      X        X
X.......GX      X        X
XXXXXXXXXX      XXXXXXXXXX
```



X	X	X	X	X	X	X	X	X	X
X	S								X
X	.								X
X	.								X
X	.								X
X	.								X
X	.								X
X	.								X
X	.	.	.	.	.	.	.	G	X
X	X	X	X	X	X	X	X	X	X

Both the straight-line path and the actual solution length have an effect on the run time of the Breadth First Search algorithm. The farther the S is from the G the more vertices and edges are ultimately queued and checked in order to find a solution. Consider the pictures below, we started at S and spread through the entire set of Vertices until the G was found. The graph to the right both the path taken and the straight distance are shorter, therefore less vertices are checked.

X	X	X	X	X	X	X	X	X	X
X	S	🟦	🟦	🟦	🟦	🟦	🟦	🟦	X
X	.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	X
X	.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	X
X	.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	X
X	.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	X
X	.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	X
X	.	🟦	🟦	🟦	🟦	🟦	🟦	🟦	X
X	.	.	.	.	.	.	.	G	X
X	X	X	X	X	X	X	X	X	X

X	X	X	X	X	X	X	X	X	X
X	S	X	.	G					X
X	.	X	.	🟦					X
X	.	.	.	🟦					X
X	🟦	🟦	🟦	🟦					X
X									X
X									X
X									X
X									X
X	X	X	X	X	X	X	X	X	X

5. For breadth first search, our solution is only storing up to 4 neighbors. We do not store a neighbor if the neighbor's data is equal to an X. So, for us more walls would mean less neighbors, and a shorter run time. Therefore, worst case scenario would be a graph with only boundaries and no walls, in which the S is as far from the G as possible. Something like this.

X	X	X	X	X	X	X	X	X	X
X	S								X
X									X
X									X
X									X
X									X
X									X
X									X
X								G	X
X	X	X	X	X	X	X	X	X	X

This would mean that each Node has between 2 and 4 neighbors and we would need to check each node(vertex) + their neighbors(edges). So, big-oh notation would be O(V+E) in worst case. The more edges there are, the higher the runtime.

6. We spent about 9 – 10 hours on this assignment in all.