Intro to Artificial Intelligence

Project 1 Report & Answers

07/15/2017

Murtala Aliyu

Ofomezie Emelle

1.

By algorithm:

a. DFS: 900 x 900. *p* value does not really drastically affect how reasonable a return time is, but the algorithm cannot reach up to 1000 x 1000 even with a p of 0.
b. BFS: 60 x 60 for any p value is a success. Likely having to do with the fact that most of the board is searched, but our BFS algorithm cannot reach 61 x 61 even with a p of 0.
c. A*Euc: 700 x 700. *p* value does not really drastically affect how reasonable a return time is, but the algorithm cannot reach up to 800 x 800 even with a p of 0.
d. A*Man: 900 x 900. *p* value does not really drastically affect how reasonable a return time is, but the algorithm cannot reach up to 1000 x 1000 even with a p of 0.
e. One extra thing to note about this question is that for all of these algorithms except for BFS, the algorithms have a chance at stack overflow error (basically the code could get an overflow error) when 0.2 >= p >= 0.32 (only for a very large dim). This is likely because there is indeed a path when p is in that range, but the path is so convoluted that it stores too much information and the program runs out of memory.

2. In this problem we pick the size of our grid to be between 2 and 60. We pick 2 as the floor because that is the smallest map that can give us a visual answer, and 60 as the ceiling because it is the highest that BFS can expand in our algorithm. Also, since we're using the same map size for all the algorithms it makes sense to cap at 60 due to BFS's shortcomings with map size.
   - Random map size generated: 25 x 25
   - Probability of a blocked cell: p = 0.2

   DFS:

```
Choose a search method...   (dfs, bfs, a*euc, a*man)
dfs
Path found!!!
S  1  B  1  1  1  1  0  B  1  1  1  1  1  B  B  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  0  0  1  B  1  1  1  B  1  1  1  B  1  1  1  1  1  1  1  1  1
1  B  1  1  1  1  1  0  B  1  1  1  1  1  1  1  1  1  1  1  1  1  1  B  1  1  1  1
1  1  B  1  B  B  1  0  0  1  B  1  1  1  B  B  1  B  1  1  B  B  1  B  B  1  1  1
1  B  1  1  1  1  1  B  0  1  B  B  0  B  B  B  1  1  1  1  1  B  B  B  1  1  1  1
1  1  1  1  1  B  1  B  0  1  1  1  B  1  1  B  1  1  B  1  B  1  1  1  1  1  1  1
1  1  B  1  1  1  1  1  B  1  1  1  B  B  1  1  1  1  1  1  1  1  1  B  1  1  1  B
1  1  1  1  1  1  1  1  1  1  1  1  1  B  1  1  0  B  1  1  B  1  1  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1  1  B  1  B  1  1  1  0  B  1  1  1  1  1  1  1
1  B  1  1  1  1  1  1  1  1  B  1  1  1  1  1  B  1  1  1  B  1  1  1  1  1  1  B
1  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  B  1  1  B  1  1  1
1  B  1  B  1  1  1  1  B  1  1  1  1  1  B  1  1  B  1  1  1  1  B  B  B  B  1  B
1  1  1  1  1  1  1  B  B  1  0  1  1  1  1  1  1  1  1  1  1  1  1  1  B  1  1  B
1  1  0  B  1  1  B  B  1  1  0  1  B  1  1  B  1  1  1  B  1  B  1  1  1  1  1  0
1  1  B  1  B  1  B  B  1  0  0  1  1  1  B  1  1  1  1  1  1  1  1  1  B  1  1  0
1  1  B  1  1  1  1  1  1  B  0  B  1  1  1  1  1  1  1  1  1  1  1  1  1  B  1  0
1  1  1  1  B  1  B  1  1  1  B  0  1  1  1  B  1  1  1  1  1  B  1  B  1  1  1  B
1  1  1  B  1  1  1  1  1  1  0  B  1  1  1  1  1  1  1  1  1  B  1  B  1  1  1  1
1  1  1  1  1  1  1  1  B  1  0  0  1  B  1  1  1  1  1  0  1  B  1  1  1  1  1  1
1  1  B  1  B  1  1  1  1  1  B  0  1  B  1  B  1  1  1  0  1  B  1  1  B  1  1  1
1  1  1  B  1  1  1  B  1  1  1  B  1  1  1  1  1  1  1  0  0  1  B  1  1  1  1  B  1
1  1  1  1  B  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  B  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  1  B  1  1  1  1  B  1  1  B  1  B  1  1  1  1  1  1  1  1  1
B  1  1  1  1  B  1  1  0  B  B  1  1  1  1  1  1  1  B  1  1  B  1  1  1  1  1  1  1
1  1  1  B  1  B  1  1  B  1  1  1  1  1  1  B  1  1  0  1  1  1  1  B  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  B  1  B  1  1  B  1  1  1  1
B  1  B  1  1  B  1  1  1  B  1  1  1  B  1  1  1  1  1  1  B  1  1  1  1  1  1  1
0  B  1  1  1  0  B  1  1  B  1  1  1  1  1  1  1  1  B  1  1  1  1  1  1  B  B  G
```

   BFS

```
bfs
Path found!!!
S  1  B  1  1  1  1  1  1  1  1  1  1  1  B  1  1  B  1  B  1  B  B  1  B
1  1  1  B  1  1  1  B  1  1  1  1  1  1  B  1  1  1  1  1  1  B  B  1  1
1  1  1  1  1  1  1  B  1  1  1  1  1  B  1  1  1  1  1  1  1  B  B  1  1
B  1  1  1  1  1  1  1  1  1  1  1  1  1  B  B  1  B  1  1  B  B  1  B  1
1  1  1  B  1  1  1  1  1  1  1  B  B  B  1  B  1  1  1  B  B  1  1  1  B  1
1  B  1  1  1  1  B  1  1  1  B  B  1  1  1  1  B  1  1  1  1  1  1  1  1  1
B  1  1  1  1  1  1  1  1  B  B  1  1  1  B  1  1  1  1  1  B  1  B  B  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1  B  1  1  1  B  B  1  B  1  B  B
1  1  1  1  1  1  1  1  1  1  1  1  1  1  B  1  1  B  1  1  1  1  B  1  1  1
B  1  1  1  1  1  B  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  B  1  1  1
1  1  1  1  1  1  1  1  1  B  1  1  B  1  1  1  1  B  B  1  1  B  1  1
1  B  B  1  1  1  1  1  B  1  1  1  1  1  1  1  1  1  B  B  1  1  B  1  1
B  B  1  B  1  1  B  1  1  B  1  1  1  1  1  1  1  1  1  1  1  B  1  1  1
1  1  1  B  B  1  B  1  B  1  1  1  1  1  1  1  1  B  1  B  1  1  1  B  1
B  1  1  1  1  B  1  1  B  1  1  1  1  B  1  1  1  B  1  B  1  B  1  B  B
1  1  B  1  1  1  1  1  1  1  B  1  1  1  B  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  B  1  1  B  0  B  1  1  1  1  B  B  1  B  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  B  B  1  1  1  B  1  1  1  1  1  1  1  1  B
B  1  1  1  1  1  1  B  1  B  1  1  1  1  1  B  1  1  1  1  1  B  B  1  1
1  1  1  1  1  1  1  B  1  1  1  B  1  1  1  1  1  1  1  1  1  1  1  1  B
1  1  B  1  B  1  1  1  1  1  1  1  1  1  B  1  1  B  1  1  1  1  1  1  1
1  1  B  B  0  B  1  1  1  1  1  1  1  1  1  1  1  1  B  1  1  1  1  1  1
1  1  B  0  0  B  B  1  1  1  1  1  B  1  1  1  1  1  1  1  B  1  1  1  1
1  1  B  B  0  B  B  1  1  1  1  1  1  1  B  1  B  1  1  1  1  1  1  1  1
1  1  1  1  B  0  0  B  1  B  B  1  B  1  1  1  1  1  1  1  1  1  1  1  G
```

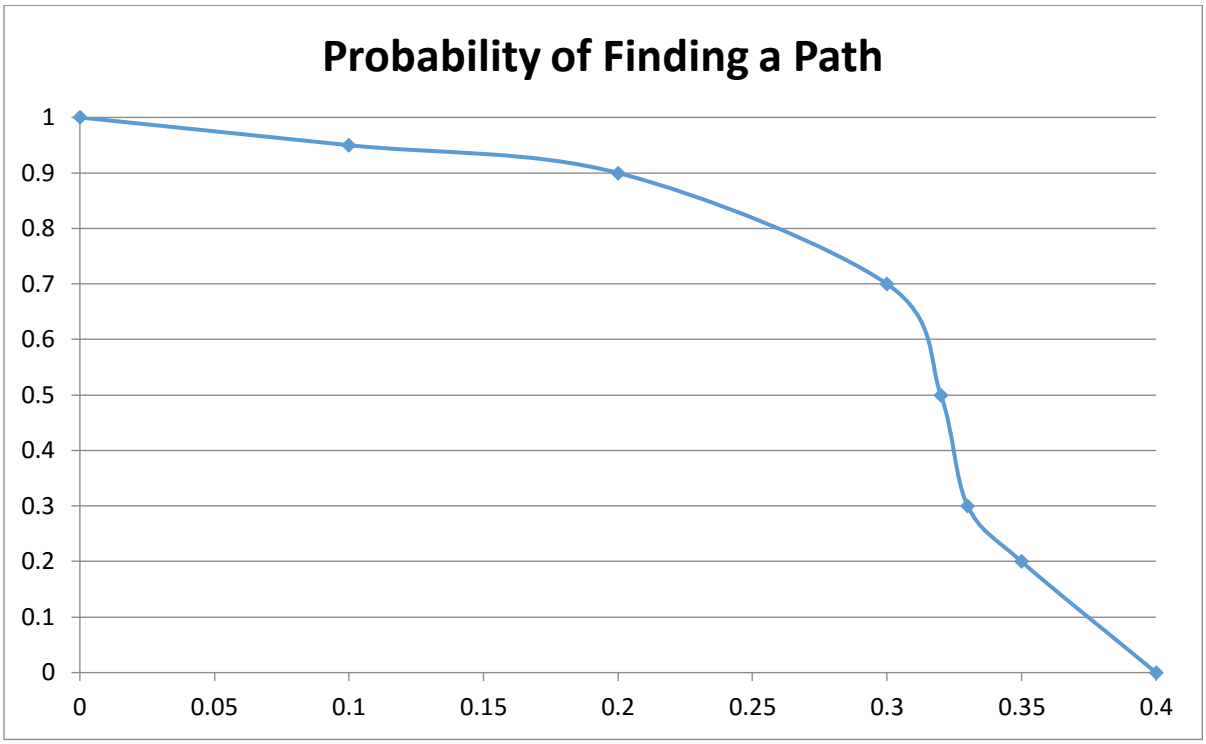
A*euc:

```
a*euc
Path found!!!
S  1  0  B  0  0  0  B  0  0  B  0  B  0  0  0  0  B  B  0  0  B  0  0  0
0  1  1  1  B  0  0  B  B  0  0  0  B  0  0  0  0  0  0  B  0  0  0  0  B
0  1  B  B  0  B  0  B  B  0  0  0  0  0  0  0  0  0  0  0  0  0  0  B  0
0  1  1  1  1  0  0  B  0  0  0  B  B  0  0  0  0  0  0  0  0  0  0  0  0
B  0  0  0  1  1  0  0  0  0  0  0  0  B  0  B  0  0  0  0  B  B  0  0  0
0  0  0  0  0  1  1  B  0  0  B  0  0  0  0  0  B  0  B  B  0  0  0  0  0
0  0  0  B  B  0  1  1  B  0  B  B  0  0  0  0  0  0  0  0  0  0  0  B  B
0  B  0  0  0  0  0  1  1  0  B  0  0  0  0  0  0  0  0  B  0  0  B  0  0
0  0  0  0  B  0  0  0  1  1  1  1  0  0  0  0  0  0  0  B  0  0  0  B  0
0  0  B  0  0  0  0  0  B  B  B  1  0  0  0  0  B  0  0  0  B  B  0  B  0
0  0  0  0  0  0  0  0  0  0  0  1  0  0  B  0  0  B  0  B  0  0  0  B  0
0  B  0  B  B  0  0  0  0  0  0  1  B  0  0  B  0  0  B  0  0  0  0  0  0
0  0  0  0  0  0  0  0  B  0  0  1  1  1  B  0  0  0  0  0  0  0  0  B  0
0  0  0  0  0  B  0  0  0  B  0  0  B  1  1  B  0  0  0  B  0  0  0  B  0
0  0  0  0  0  0  0  0  0  0  0  B  B  B  0  1  1  0  0  0  0  0  B  0  0  B
B  0  0  B  0  0  0  0  0  0  0  0  0  0  0  0  1  1  B  0  B  B  0  0  B  0
0  0  0  0  0  B  0  0  0  0  B  B  0  0  B  0  1  1  1  1  1  0  B  B  0
0  0  0  0  0  0  0  0  0  0  0  0  0  B  0  B  0  1  1  B  1  1  0  0  0
0  B  0  0  0  B  0  0  B  0  0  0  0  0  0  0  0  B  B  B  B  1  B  0  0
0  0  0  0  0  0  B  0  0  0  0  0  0  0  0  0  0  B  B  0  0  1  B  0  B
0  0  0  0  0  0  0  0  B  0  0  B  0  0  0  0  0  0  0  0  0  1  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  B  B  B  0  B  0  0  1  B  B  0
0  B  0  0  B  0  B  0  0  0  0  B  B  0  0  0  0  0  0  0  0  1  1  B  0
B  0  0  0  0  0  0  0  0  0  0  B  0  0  0  B  0  0  B  0  0  0  0  1  1
0  0  0  0  0  0  0  B  0  B  0  B  0  0  0  0  0  0  0  0  0  0  0  B  B  G
```

A*man:

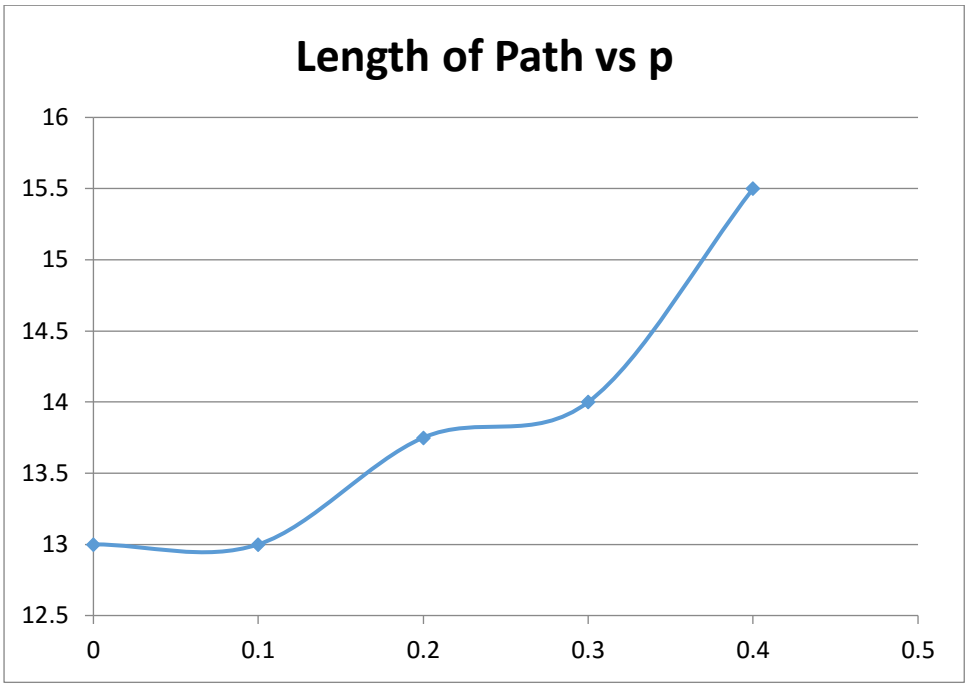
```
a*man
Path found!!!
S  1  1  1  1  1  1  1  1  1  B  0  0  B  0  0  0  B  0  0  0  0  0  0  B
0  0  B  0  0  B  0  0  0  1  1  1  B  0  B  B  B  0  B  0  0  0  0  0  B
0  0  0  0  0  0  0  0  0  0  B  1  1  1  1  1  1  1  1  1  1  1  1  1
0  0  0  0  0  0  0  B  0  B  0  0  0  0  0  0  0  0  0  0  0  0  1  1
0  0  0  0  0  0  B  0  0  0  0  0  0  0  0  0  B  0  0  B  0  0  B  1  1
0  0  0  B  0  0  B  0  B  0  0  0  B  0  0  0  0  0  0  0  0  1  1  B  1
0  0  0  B  0  0  0  B  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1
0  0  0  0  B  0  0  0  0  0  0  0  0  0  B  0  0  0  0  1  1  1  B  1  1
0  0  0  B  0  0  0  B  0  0  0  0  B  0  0  0  0  B  B  1  B  B  0  B  1
0  0  0  B  0  0  0  B  0  0  0  0  0  0  B  B  B  B  0  1  1  1  1  1  B
0  B  0  0  0  B  0  0  0  0  0  0  B  0  0  0  0  0  0  0  0  0  0  1  1
0  0  B  0  B  0  B  0  0  0  0  0  0  B  0  B  0  B  B  0  0  0  0  1  1
0  0  0  0  0  0  B  0  0  0  0  0  B  0  0  0  0  0  0  0  0  0  0  1  B
0  0  0  0  0  0  0  0  0  0  0  0  0  0  B  0  0  0  0  0  0  0  0  1  1
0  B  B  B  0  0  0  0  0  0  B  0  0  0  B  0  0  0  0  B  B  0  0  0  1
0  0  0  0  0  0  0  B  0  0  0  0  B  0  0  0  0  B  0  0  0  B  0  B  1
0  0  B  0  0  0  0  0  0  0  0  0  0  B  B  0  0  0  0  0  B  0  0  1  1
0  0  B  B  0  0  0  0  0  0  0  B  0  0  0  B  0  0  0  0  0  0  1  1  1
0  0  B  0  0  0  0  0  0  0  0  B  0  0  0  0  0  0  0  0  0  0  1  B  1
B  0  0  0  0  0  0  0  0  0  0  0  0  B  0  0  0  0  B  0  B  1  1  1  B
0  0  0  0  0  B  0  B  B  0  0  B  B  0  0  0  0  0  0  0  0  1  B  1  1
0  0  0  0  0  0  0  B  0  0  0  B  0  B  0  0  0  B  B  0  0  1  B  B  B
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  B  1  1  B
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  B  0  0  0  0  0  0  1  1
0  0  0  0  0  B  0  0  0  B  0  0  0  B  0  0  0  0  0  0  0  0  0  0  G
```

3.



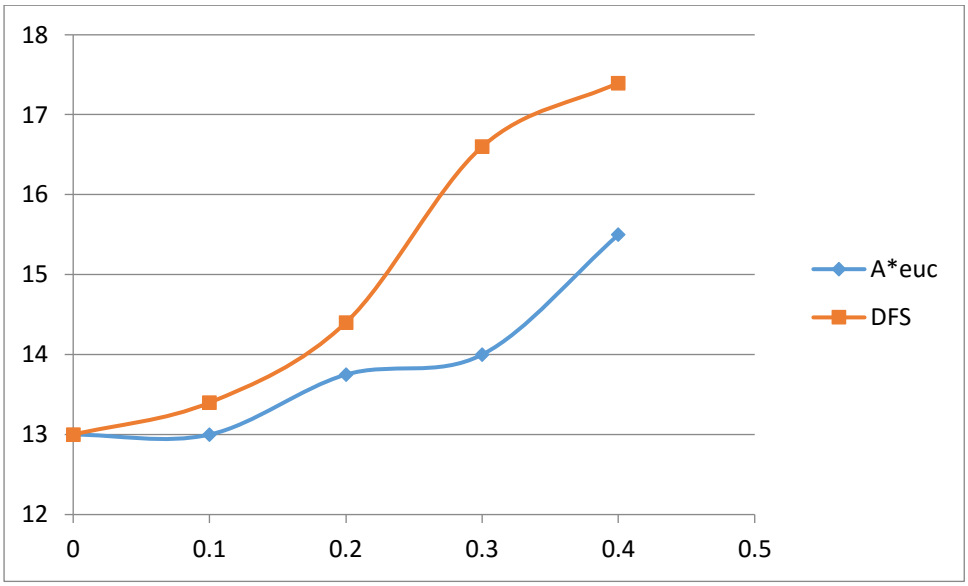

The x-axis is *p,* the y-axis is the probability of finding a path at the given *p*. Specifically the point where the most drop off in probability is when p = 0.32. We cut off the graph where p > 0.4 because there were never any paths that were formed after p = 0.4 at all. The best algorithm for this is actually BFS in my opinion. We are only concerned with whether or not there is a clear path given a range of p values, not what the shortest path is. Therefore since BFS is always a complete algorithm it can return just true (path exists) or false (path does not exist).

4.




7x7 dim. A* is the most useful path finding algorithm here as we are trying to find the shortest path between the start and the goal. Also, as stated in answer 3, after p = 0.32 there's a significant drop off in whether or not a path exists for a given dim and reduces to effectively 0% at p = 0.4, so there wasn't any data past p = 0.4.

5.



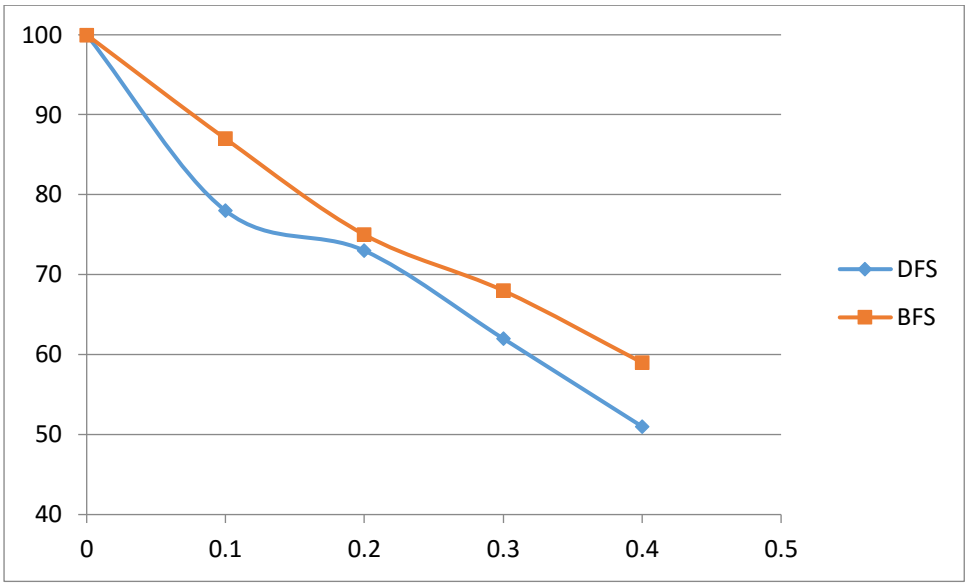

7x7 dim. x-axis is p, y-axis is the average length of the path generated. As one would expect, the paths generated by A* are smaller than the paths generated by DFS, and if say the dim was increased, the difference between these two lines as p increases would be even greater.

6. Size of map: 30 x 30
- Average: 10 tries for each p value

| | A*euc | A*man |
|---|---|---|
| p = 0.0 | 115 | 114 |
| p = 0.05 | 115.3 | 121.3 |
| p = 0.10 | 114 | 128.7 |
| p = 0.15 | 113.7 | 134 |
| p = 0.20 | 115.7 | 140.9 |
| p = 0.25 | 123.1 | 140.5 |
| p = 0.30 | 121.5 | 126.7 |
| p = 0.35 | 164.8 | 128.1 |
| p = p0 = 0.40 | 165.8 | 151.1 |
| p = 0.45 | | |
| p = 0.50 | | |

*These statistics are valid ONLY for maps that have a path from Start to Goal*

7.




10x10 dim. X-axis is value *p* and y-axis is average number of nodes expanded. As expected, DFS expands fewer nodes than BFS since DFS completes its algorithm by expanding all the children of the current node before looking at any neighbors, while BFS does not get to the last row of nodes until the very end after checking each row of a tree. BFS is a complete algorithm, while DFS could save you some time and memory. ***SECOND PART OF THIS QUESTION STILL NEEDS TO BE ANSWERED USING ANSWER FOR #6****

8.

Bonus #1: There was no point asking us to implement UFCS when the cost of moving from one node to another in this assignment is always 1 already. Essentially the code for BFS already covers UFCS.

Bonus#2 (*only if you feel like it*): Hi