# Question 6

The actual code has further comments on the implementation of the BFS, DFS and A\* search. The cost for the BFS and DFS is equivalent to the nodes explored value. The 999 view represents the black wall. The path goes from 0 to the end of the path (nodesexplored in BFS, A\* search and mindistance in DFS)

## BFS

The BFS implementation uses a queue for the coordinates to check out. We start from the starting coordinates and add all the neighbors of it to the back of the queue. We then add the starting coordinates to the visited queue (already populated with the black squares in the maze). From there we keep repeating this till we get a coordinate that equals the ending coordinate. Whatever coordinate was added first will be the first ones to be analyzed this replicating the FIFO methodology and mimicking Breadth First Search.

The output of this function is as follows:

### S to E1

Text

Description automatically generated

Maze Output

A picture containing calendar

Description automatically generated

### Text Description automatically generatedS to E2

A picture containing calendar

Description automatically generatedMaze Output

### (0,0) to (24,24)

A picture containing text

Description automatically generated

A picture containing calendar

Description automatically generatedMaze Output

## DFS

The Depth First Search uses a stack in order to store nodes. These nodes contain the nodes coordinates and the parent node. Similar to the BFS implementation we first create a starting node with the coordinates and populate it onto the stack. We then pop it off the stack and see if it is the ending coordinate. If not, we append to the visited list and add its neighbors to the TOP of the stack. We then repeat till we find the ending node.

Once we find the ending node, we then work backwards to find the path back to the start. Therefore, we get the path generated from the DFS and the min distance.

### S to E1

A picture containing text

Description automatically generated

A picture containing calendar

Description automatically generatedMaze Output

### Text Description automatically generatedS to E2

A picture containing table

Description automatically generatedMaze Output

### Text Description automatically generated(0,0) to (24,24)

A picture containing calendar

Description automatically generatedMaze Output

## A\* Search

The A\* search is very similar to BFS except I created a heuristic function to change the cost of analyzing the node. I chose to make my heuristic function add additional cost for values that do not fall in between the ranges of the startingX and the endingY or the startingX and the endingY.

So cost of the node is valuated as follows:

1 + (1 if startingX < nodeX < endX is not true) + (1 if startingY < nodeY < endY is not true)

I then sort the list of nodes to have the list sorted by the cost, so we look at the node with the lower cost first. In our output we got less nodes expanded than our BFS, so it seems to be successful!

### Text Description automatically generatedS to E1

Table

Description automatically generated with medium confidenceMaze output

### Text Description automatically generatedS to E2

Table

Description automatically generatedMaze Output

### A picture containing background pattern Description automatically generated(0,0) to (24,24)

A picture containing calendar

Description automatically generatedMaze Output