A+ Computer Science	Graphs – THE MAZE	Lab Value - 100

Lab Goal : This lab was designed to teach you more about recursion.

Lab Description : A maze is a two dimensional structure with an entrance and sometimes an exit. Determine if any maze has an exit. All of the mazes will have the entrance point at the top left corner. The exit point can be anywhere in the far right column. A valid path will be a continuous block of 1s that connect the top left corner to any spot on the far right column. A valid path can only be connected horizontally or vertically. Diagonal connections are not legal. The 1s in the file represent the path and 0s represent the walls of the maze. Output `exit found` if there is a path leading to an exit out of the maze. Output `exit not found` if there is not a path leading out of the maze.

Helpful Hints / Assumptions: All input will be 1s and 0s. The 1s represent the path and 0s show that there is no path in that area.

Sample Output :

```
1 0 0 0 1
1 1 1 1 0
0 0 1 0 1
0 1 1 1 0
0 0 0 0 1
exit not found

1 0 0 0 0 1 1
1 1 1 1 0 1 0
0 0 1 0 0 1 0
0 1 1 1 0 1 0
0 1 0 1 0 1 0
0 1 0 1 1 1 0
0 1 0 1 0 0 1
exit found

1 0 0 0 0 1 0
1 1 1 1 0 1 0
0 0 1 0 0 1 0
0 1 1 1 0 1 0
0 1 0 1 0 1 0
0 1 0 1 1 1 0
0 1 0 1 0 1 0
exit not found

1 0 1 1 0 1 0
1 1 1 1 1 1 0
0 0 1 0 0 0 1
0 1 1 1 1 1 1
0 1 0 1 0 1 0
1 1 1 1 1 1 0
0 1 0 1 0 1 0
exit found

1 0 1 1 0 1 0 1 1 1
1 1 1 1 1 1 0 1 0 1
0 0 1 0 0 0 1 1 1 1
0 1 1 1 1 1 1 1 0 1
0 1 0 1 0 1 0 1 0 1
1 1 1 1 1 1 0 1 1 1
0 1 0 1 0 1 0 0 0 1
0 1 1 1 0 1 0 0 0 0
0 1 0 1 0 1 0 1 1 1
0 1 1 1 1 1 0 1 1 1
exit found

1 0 1 1 0 1 1 1 1 0
1 1 1 1 1 1 0 1 0 1
0 0 1 0 0 0 1 1 1 0
0 1 1 1 1 1 1 1 0 1
0 1 0 1 0 1 0 1 0 1
1 1 1 1 1 1 0 1 1 0
0 1 0 1 0 1 0 0 0 0
0 1 1 1 0 1 0 0 1 1
0 1 0 1 0 1 0 1 1 1
0 1 1 1 1 1 0 1 1 1
exit not found
```

Files Needed ::

Maze.java
MazeRunner.java

algorithm help

```
if ( r and c are in bounds and current spot is a 1 )
   mark spot as visited
   if we are at the exit
      mark exit found as true
   else
      4 recursive calls up down left right
```

Sample Data :

```
5
1 0 0 0 1
1 1 1 1 0
0 0 1 0 1
0 1 1 1 0
0 0 0 0 1
7
1 0 0 0 0 1 1
1 1 1 1 0 1 0
0 0 1 0 0 1 0
0 1 1 1 0 1 0
0 1 0 1 0 1 0
0 1 0 1 1 1 0
0 1 0 1 0 0 1
7
1 0 0 0 0 1 0
1 1 1 1 0 1 0
0 0 1 0 0 1 0
0 1 1 1 0 1 0
0 1 0 1 0 1 0
0 1 0 1 1 1 0
0 1 0 1 0 1 0
7
1 0 1 1 0 1 0
1 1 1 1 1 1 0
0 0 1 0 0 0 1
0 1 1 1 1 1 1
0 1 0 1 0 1 0
1 1 1 1 1 1 0
0 1 0 1 0 1 0
10
1 0 1 1 0 1 0 1 1 1
1 1 1 1 1 1 0 1 0 1
0 0 1 0 0 0 1 1 1 1
0 1 1 1 1 1 1 1 0 1
0 1 0 1 0 1 0 1 0 1
1 1 1 1 1 1 0 1 1 1
0 1 0 1 0 1 0 0 0 1
0 1 1 1 0 1 0 0 0 0
0 1 0 1 0 1 0 1 1 1
0 1 1 1 1 1 0 1 1 1
10
1 0 1 1 0 1 1 1 1 0
1 1 1 1 1 1 0 1 0 1
0 0 1 0 0 0 1 1 1 0
0 1 1 1 1 1 1 1 0 1
0 1 0 1 0 1 0 1 0 1
1 1 1 1 1 1 0 1 1 0
0 1 0 1 0 1 0 0 0 0
0 1 1 1 0 1 0 0 1 1
0 1 0 1 0 1 0 1 1 1
```