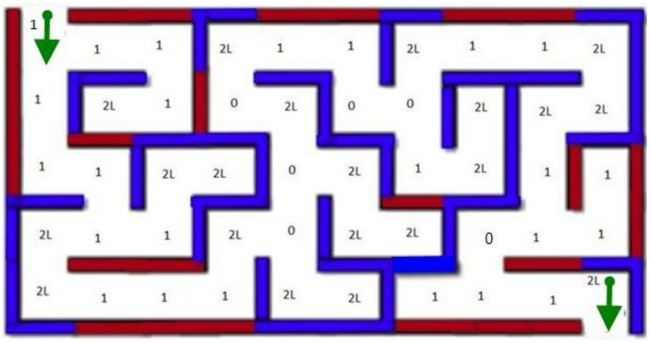
Lab - Recursion - Backtracking
Two-D Maze
25 points
See Canvas for due date



This assignment:

Solve for a path through a Two-Dimensional maze by use of exhaustive search and backtracking.

Given a two-dimensional array like the one shown below, create a recursive Java program that will traverse the array looking for a path from the upper left corner to the lower right.

```java
private int[][] grid = {  {1,1,1,0,1,1,0,0,0,1,1,1,1},
                          {1,0,1,1,1,0,1,1,1,1,0,0,1},
                          {0,0,0,0,1,0,1,0,1,0,1,0,0},
                          {1,1,1,0,1,1,1,0,1,0,1,1,1},
                          {1,0,1,0,0,0,0,1,1,1,0,0,1},
                          {1,0,1,1,1,1,1,1,0,1,1,1,1},
                          {1,0,0,0,0,0,0,0,0,0,0,0,0},
                          {1,1,1,1,1,1,1,1,1,1,1,1,1}};
```

Program specifications:

In the maze, values represented are:

- 1 – An available space for a path (an open ‘room’)
- 0 – A ‘wall’
- 3 – A visited room
- 7 – A room on a successful path

Your program should:

- Start in the upper left corner
- Attempt to move in a direction

    - Check that the ‘room’ you want to move into is inside the maze
    - Check that the room you want to move to is available (not a wall)
    - Check that the room you want to move to has not been visited previously

- If the move is not valid try to move in another direction
- If the is valid, ‘move’ into the ‘room’ and mark it as ‘visited’ (3)
- If no valid move is available, unwind in your recursive call stack to the previous call and try the next direction from there
- Repeat until either you’ve reached the goal or you’ve determined that there is no path through the maze.

Some suggestions:

- Determine what the base cases will be
- Your recursive method should return a variable of type boolean.
- Your recursive method’s signature should accept two parameters only.

To turn in

- Print your solution and submit in class.