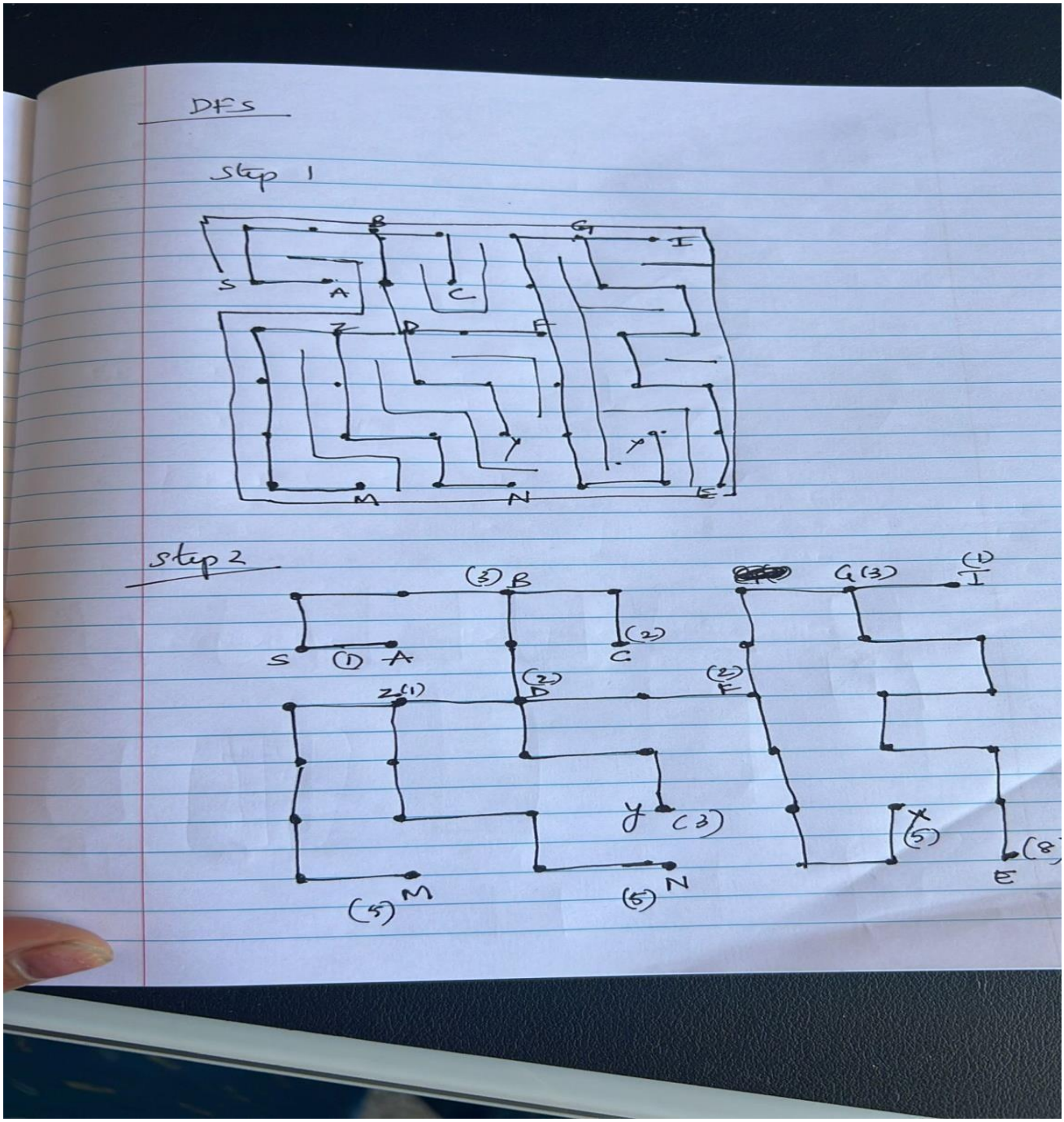
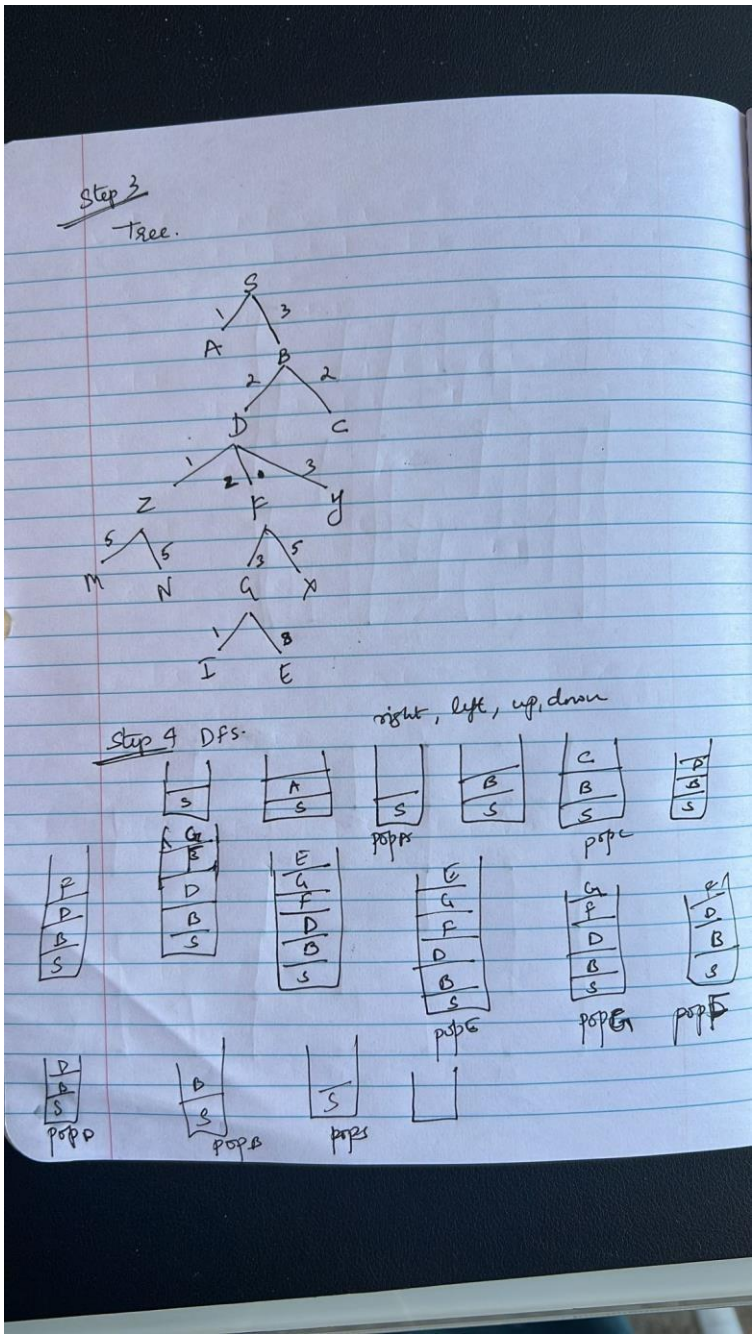
Week 11: Homework 1: Depth-First Traversal: The Maze

Q3 => Conduct Depth First Traversal (DFT) on a maze

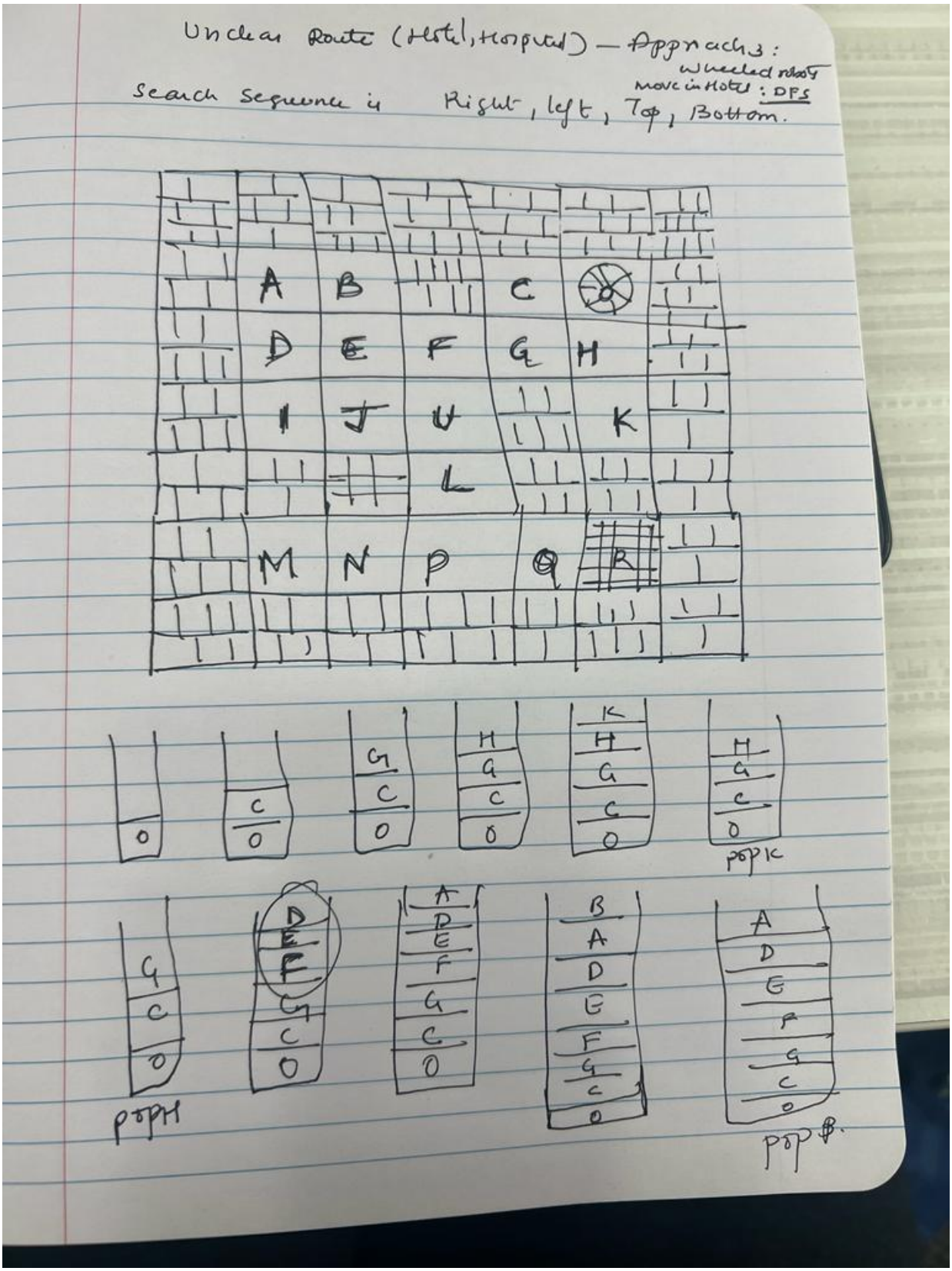


Week 11: Homework 1: Depth-First Traversal: The Maze

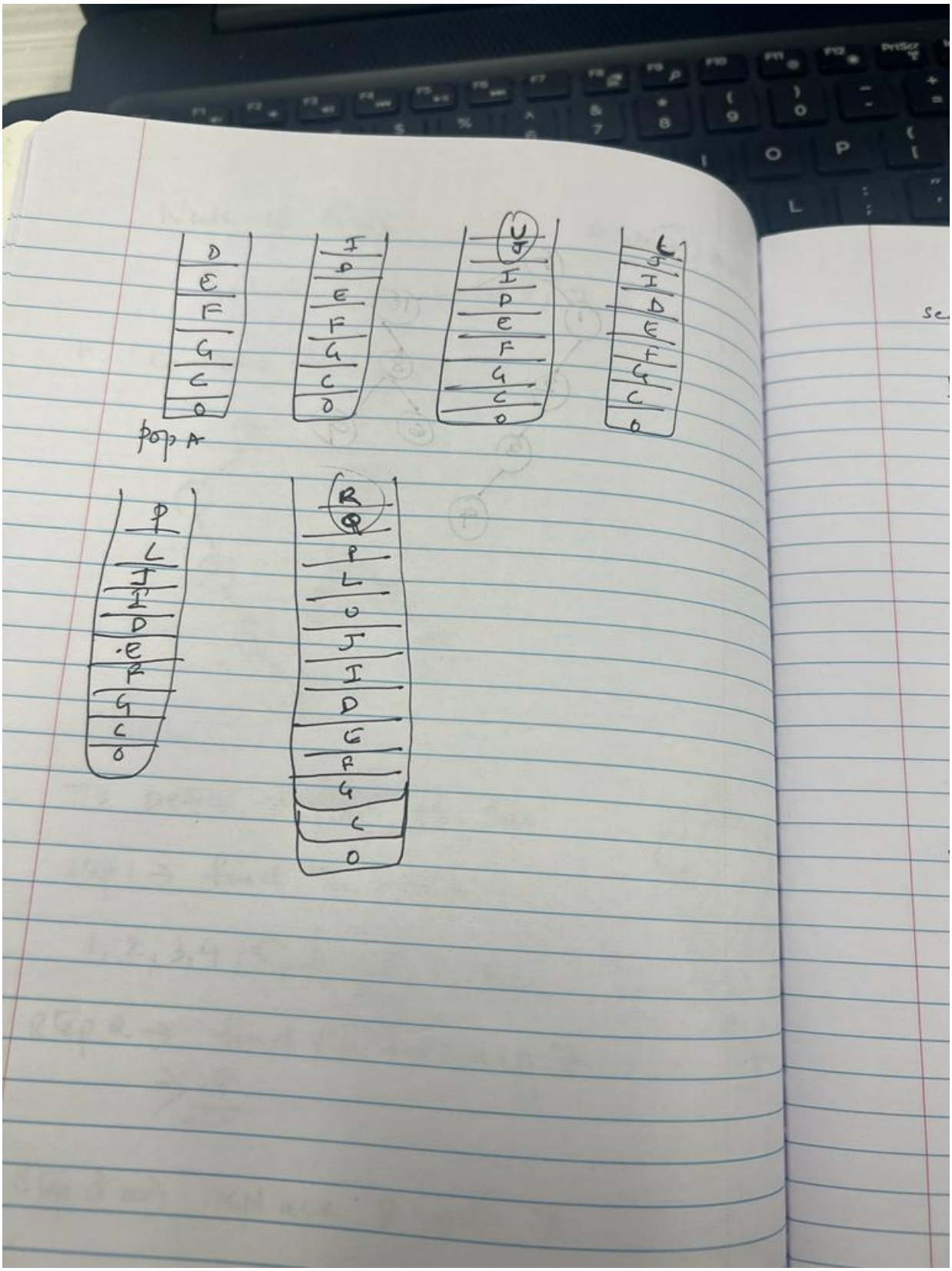


Approach 2

Week 11: Homework 1: Depth-First Traversal: The Maze



Week 11: Homework 1: Depth-First Traversal: The Maze



Week 11: Homework 1: Depth-First Traversal: The Maze

Code

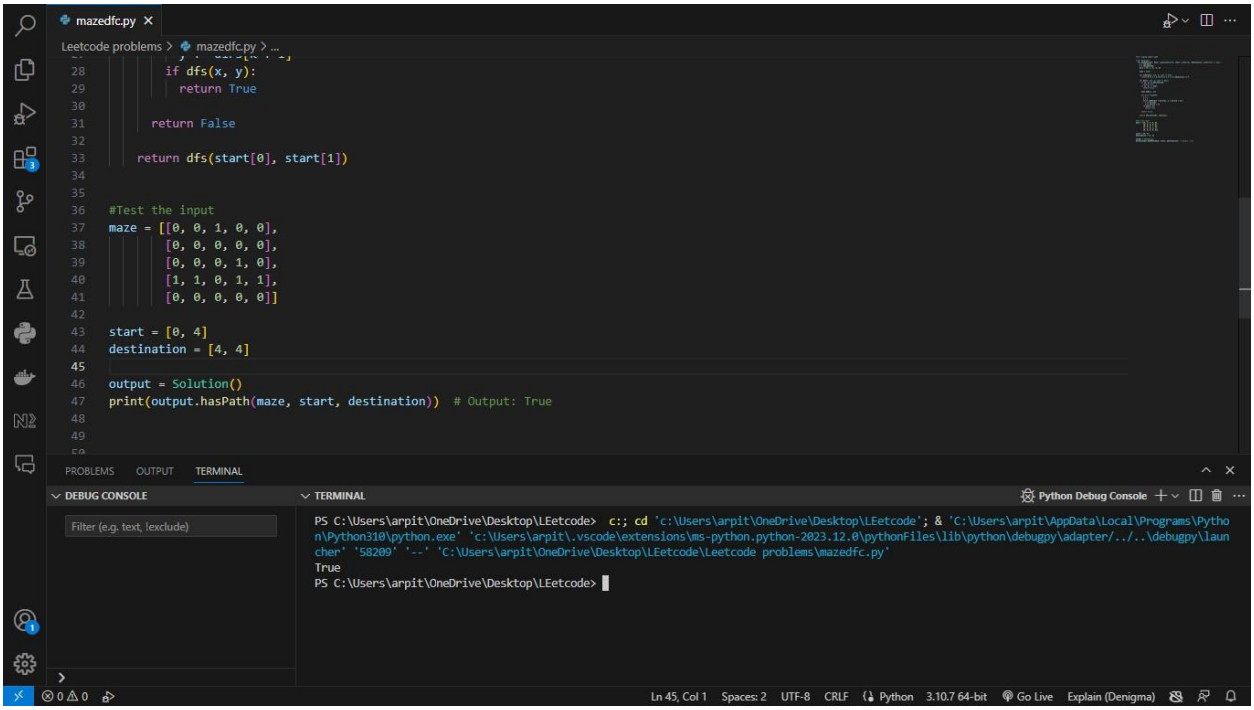
```
from  typing import List
 class Solution:   def hasPath(self, maze: List[List[int]], start:
List[int], destination: List[int]) -> bool:     m = len(maze)     n =
len(maze[0])     dirs = [0, 1, 0, -1, 0]
     seen =
set()
     def isValid(x: int, y: int) ->
bool:
      return 0 <= x < m and 0 <= y < n and maze[x][y] == 0
     def dfs(i: int, j: int) ->
bool:       if [i, j] ==
destination:
        return True
if (i, j) in seen:
        return False
       seen.add((i,
j))
       for k in
range(4):
        x = i         y = j         while isValid(x
+ dirs[k], y + dirs[k + 1]):
          x += dirs[k]
y += dirs[k + 1]
if dfs(x, y):
return True

      return False
     return dfs(start[0],
start[1])
```

Result with different test scenarios :

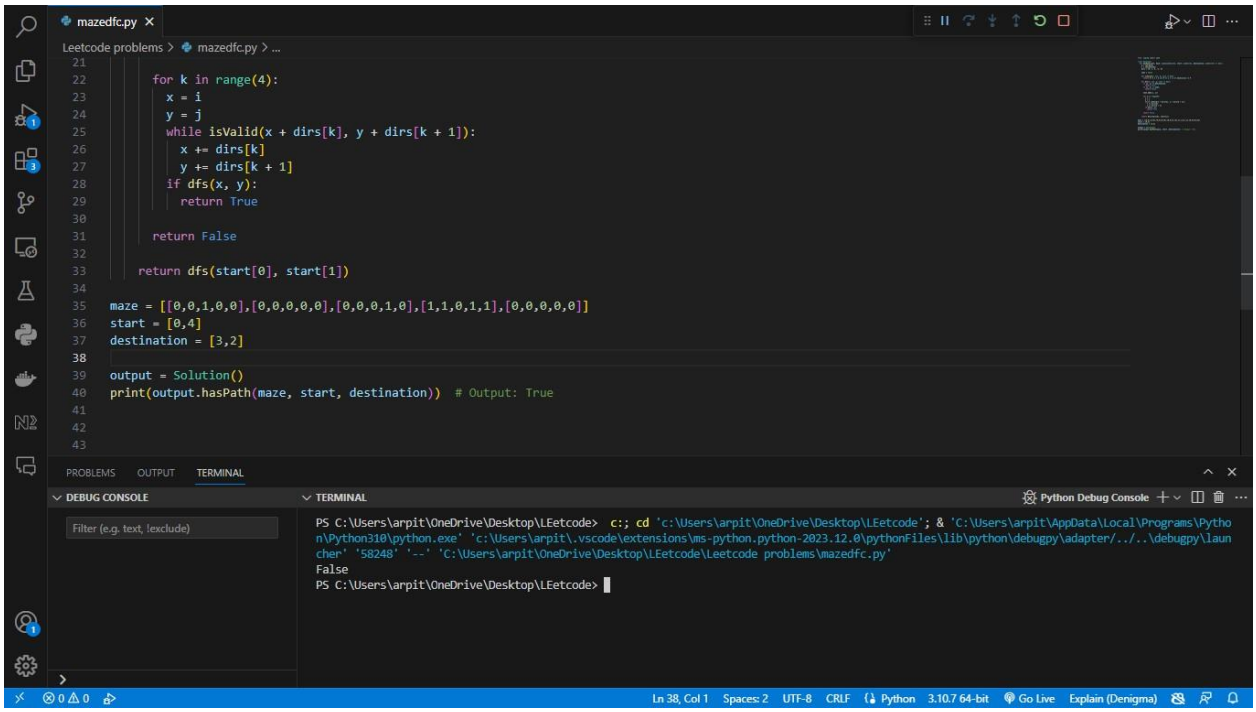
Week 11: Homework 1: Depth-First Traversal: The Maze

```
maze = [[0, 0, 1, 0, 0],
[0, 0, 0, 0, 0],
        [0, 0, 0, 1, 0],
        [1, 1, 0, 1, 1],
        [0, 0, 0, 0, 0]]
 start = [0, 4]
destination = [4,
4]
```



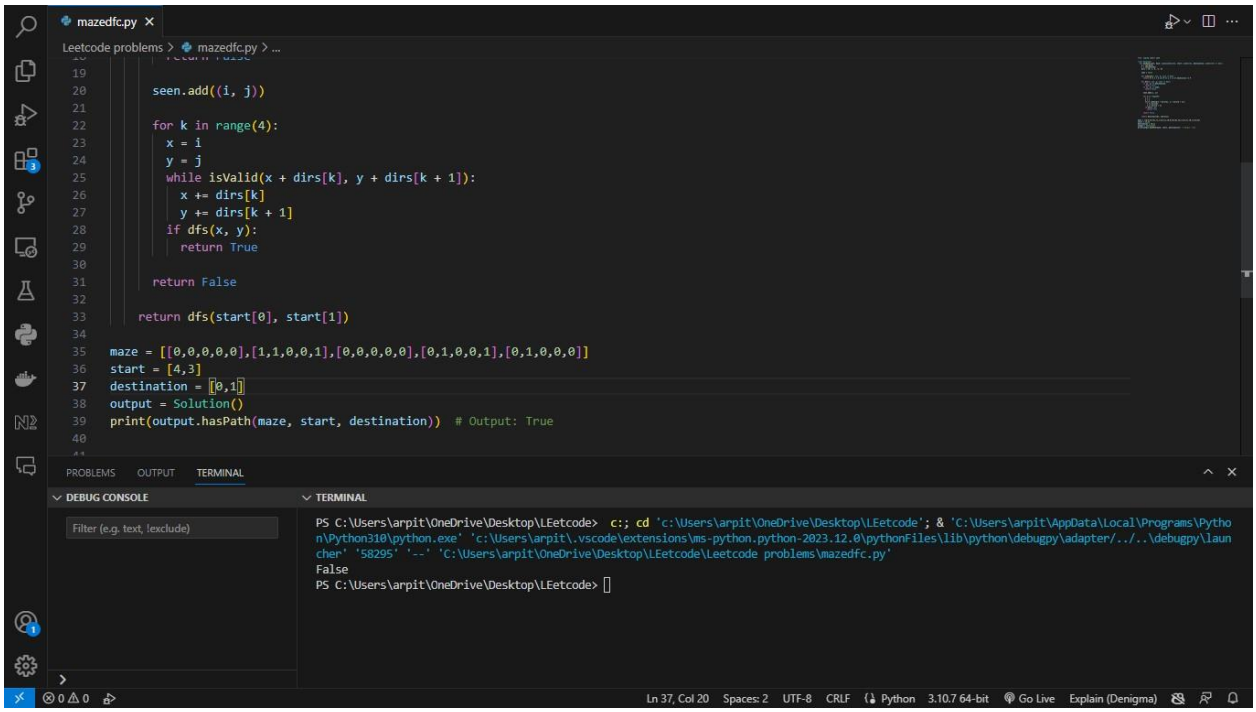

```
maze = [[0,0,1,0,0],[0,0,0,0,0],[0,0,0,1,0],[1,1,0,1,1],[0,0,0,0,0]] start =
[0,4] destination = [3,2]
```

Week 11: Homework 1: Depth-First Traversal: The Maze




```
maze = [[0,0,0,0,0],[1,1,0,0,1],[0,0,0,0,0],[0,1,0,0,1],[0,1,0,0,0]] start
= [4,3] destination = [0,1]
```

Week 11: Homework 1: Depth-First Traversal: The Maze



GitHub URL: https://github.com/jesalshah14/Maze_DFS_Project