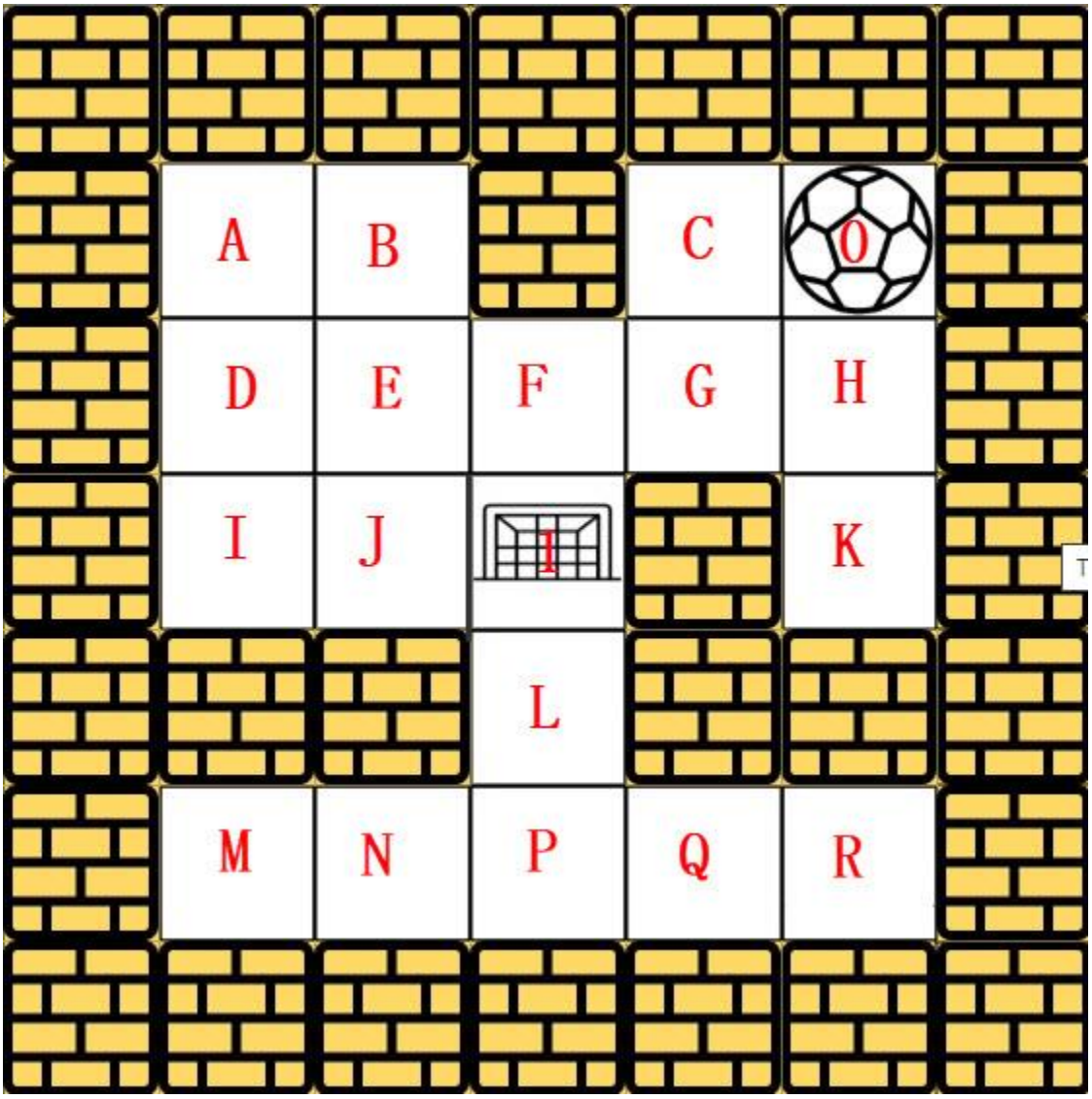
Week 11 q3



Right=>left=>top=>bottom

DFT

				K				1
			H	H	H		F	F
		G	G	G	G	G	G	G
	C	C	C	C	C	C	C	C
0	0	0	0	0	0	0	0	0

CHATGPT CODE

```
def hasPath(maze, start, destination):
    rows, cols = len(maze), len(maze[0])

    # Helper function to perform DFS
    def dfs(x, y):
        # Check if the current position is the destination
        if [x, y] == destination:
            return True

        # Mark the current position as visited
        maze[x][y] = -1

        # Explore four possible directions: up, down, left, and right
        directions = [(0, 1), (0, -1), (1, 0), (-1, 0)]
        for dx, dy in directions:
            new_x, new_y = x, y

            # Keep rolling in a direction until hitting a wall or the boundary
            while 0 <= new_x + dx < rows and 0 <= new_y + dy < cols and
maze[new_x + dx][new_y + dy] != 1:
                new_x += dx
                new_y += dy

            # If the new position is not visited, perform DFS on it
            if maze[new_x][new_y] == 0 and dfs(new_x, new_y):
                return True

        # If no path found, return False
        return False

    # Call DFS from the starting position
    return dfs(start[0], start[1])

# Test data
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]

# Check if there is a path from start to destination
```

```
print(hasPath(maze, start, destination))  # Output: True
```

TEST CASE