

Week 11: Homework 3: Project: Depth-First Traversal: The Maze

Manual Approach:

Manual Solution of the Depth-First traversal:

The Maze :-

A	B		C	0
D	E	F	G	H
I	J	1		K
		L		
M	N	O	P	Q

0 → Source
1 → Destination

Sequence of search :-
Right → Left → Top → Bottom

Diagram illustrating the search path (0 to 1) using Depth-First Traversal (Right → Left → Top → Bottom):

- 0
- C
- G
- H
- G
- D
- A
- I
- J
- 1

python Code:

```
def hasPath(maze, start, destination):
```

```
    def dfs(x, y):
```

```
        if (x, y) == destination:
```

```
            return True
```

```
if (x, y) in visited:
```

```
    return False
```

```
visited.add((x, y))
```

```
# Explore all four directions (up, down, left, right)
```

```
for dx, dy in [(-1, 0), (1, 0), (0, -1), (0, 1)]:
```

```
    nx, ny = x, y
```

```
    # Keep moving in the current direction until  
    hitting a wall or boundary
```

```
    while 0 <= nx + dx < len(maze) and 0 <= ny + dy <  
len(maze[0]) and maze[nx + dx][ny + dy] == 0:
```

```
        nx += dx
```

```
        ny += dy
```

```
    if dfs(nx, ny):
```

```
        return True
```

```
return False
```

```
visited = set()
```

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

```
# Test case
```

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

#Assumption2

```
maze3  
= [[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]]  
start3= [4,3]  
des3 =[0,1]  
print(hasPath(maze3,start3,des3))
```

Output:

True

False

Screen Shot:

