CS 501 Practical Application of Algorithm :

Student ID : 19615

Student Name : Gayatri Kolekar

Week10 Homework1: Project: Depth-First Travesal : The Maze

Question34: “490. The Maze”

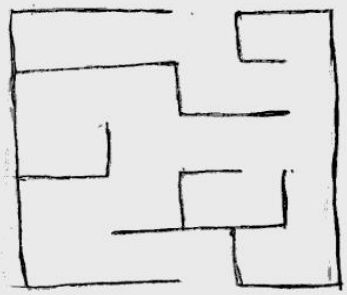
Step 1: Manual Process to Demonstrate concepts :

Step 1.1: Tree

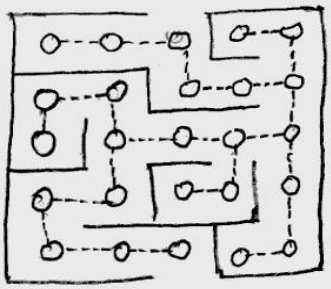
Step 1.2 : Matrix

Maze : Tree (List)

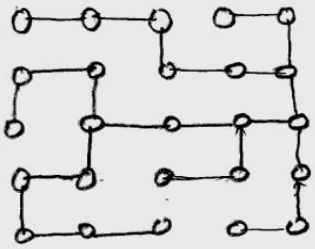
step 1:



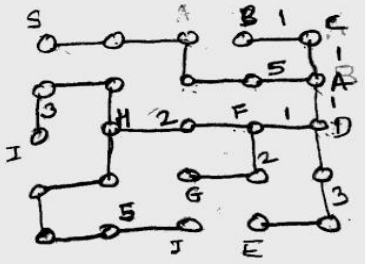
step 2:



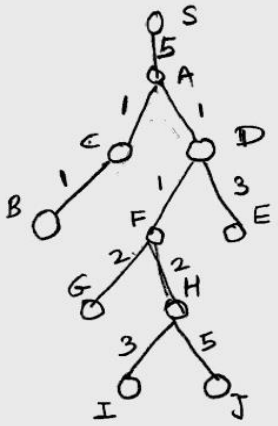
step 3:



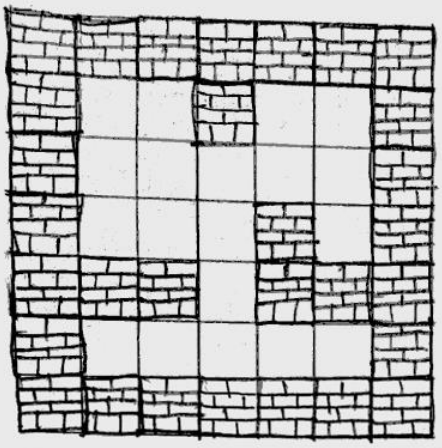
step 4:

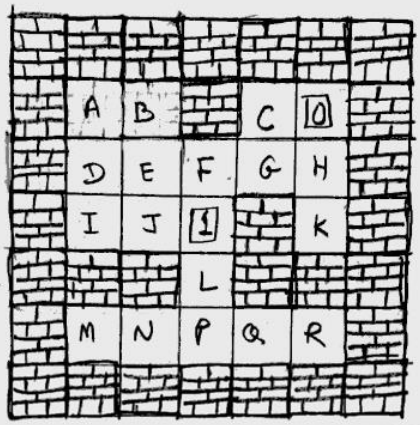


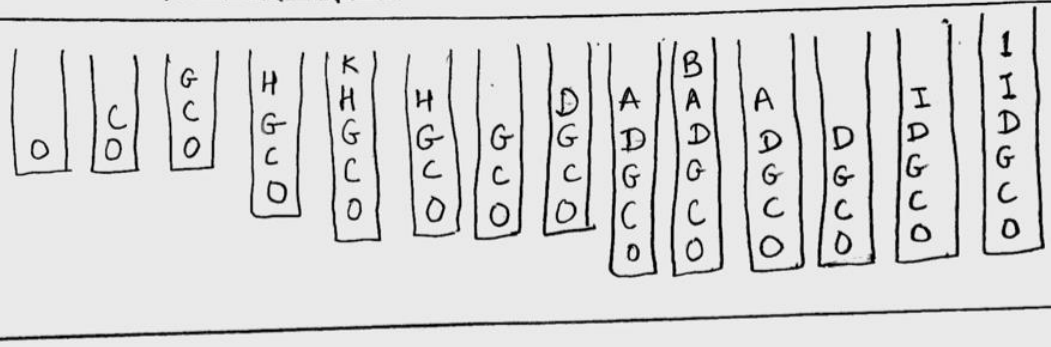
step 5:



Maze: Grid (Matrix)







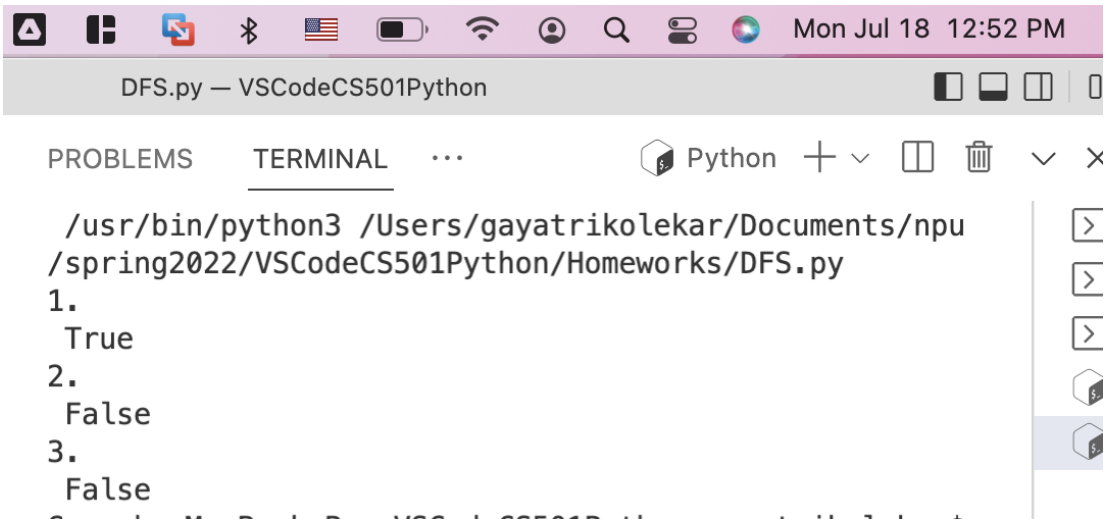
Step 2 : Implement a Python Solution using the algorithm and test the Python code :

```
from typing import List



class Solution:
    def hasPath(self, maze: List[List[int]], start: List[int], destination: List[int]) -> bool:
        '''
        Detailed explanation is available at
        https://medium.com/@edward.zhou/leet-code-53-maximum-subarray-detailed-explained-python3-solution-d91c7affc02a
        '''
        visited = []
        dirs = [(-1,0), (1,0), (0,-1), (0,1)]
        dest = (destination[0],destination[1])


        def rollFrom(pos):
            #check all possible stop positions that current pos can roll to
            #and exclude those that are already in visited
            #and then keep rolling from the rest
            #print("rolling from {}".format(pos))
            newStops = []
            for d in dirs:
                newX = pos[0]
                newY = pos[1]
                while(True): #rolling
                    possibleNewX = newX + d[0]
                    possibleNewY = newY + d[1]
                    if (possibleNewX >= 0 and possibleNewX < len(maze) ) and (possibleNewY >= 0 and possibleNewY < len(maze[0])) and (maze[possibleNewX][possibleNewY] != 1):
                        newX = possibleNewX
```

```
                    newY = possibleNewY
                    continue
                else:
                    break
            newStop = (newX, newY)
            if newStop == dest:
                return True
            newStops.append(newStop)

        visited.append(pos)

        for newStop in newStops:
            if newStop not in visited:
                if rollFrom(newStop):
                    return True
        return False

    startPos = (start[0], start[1])
    return rollFrom(startPos)

s = Solution()
print("1. ")
lst=[[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]
dest=[4, 4]
print("", s.hasPath(lst,start, dest))

print("2. ")
lst=[[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]
dest=[3, 2]
print("", s.hasPath(lst,start, dest))

print("3. ")
lst=[[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=[4, 3]
dest=[0, 1]
print("", s.hasPath(lst,start, dest))
```

Python Program Code Output :



Step 3 : Update your portfolio about the Maze project :

https://docs.google.com/presentation/d/1LBi9q1NpsYX61x5ua-cXb34JDBOIHheqmj32uZPHjlg/edit?usp=sharing

Step 4 : Submit the URL of your GitHub webpage as the homework answer.