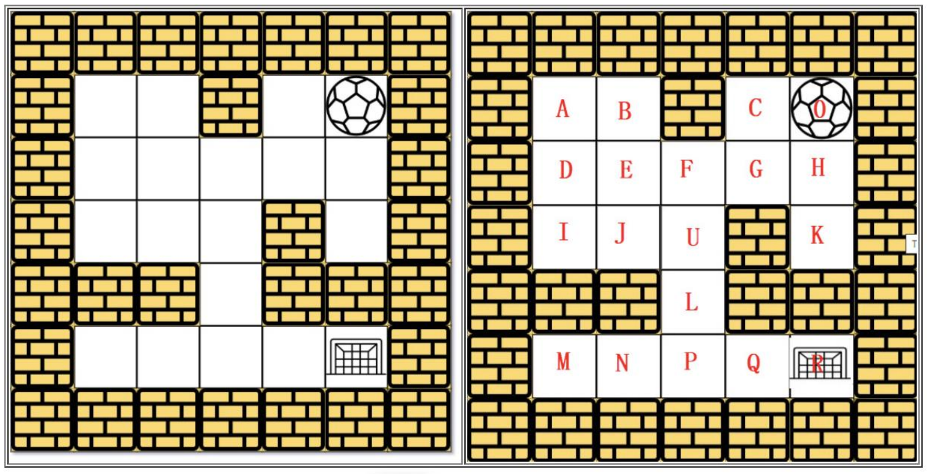
CS 501 Practical Application of Algorithm :

Student ID : 19615

Student Name : Gayatri Kolekar

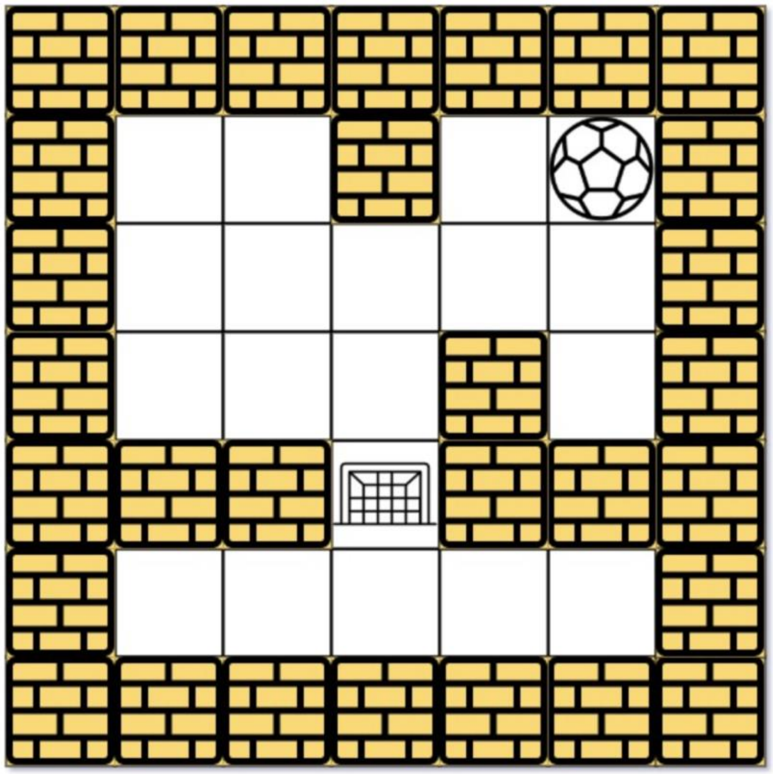
Week12: Homework1: Aug9:

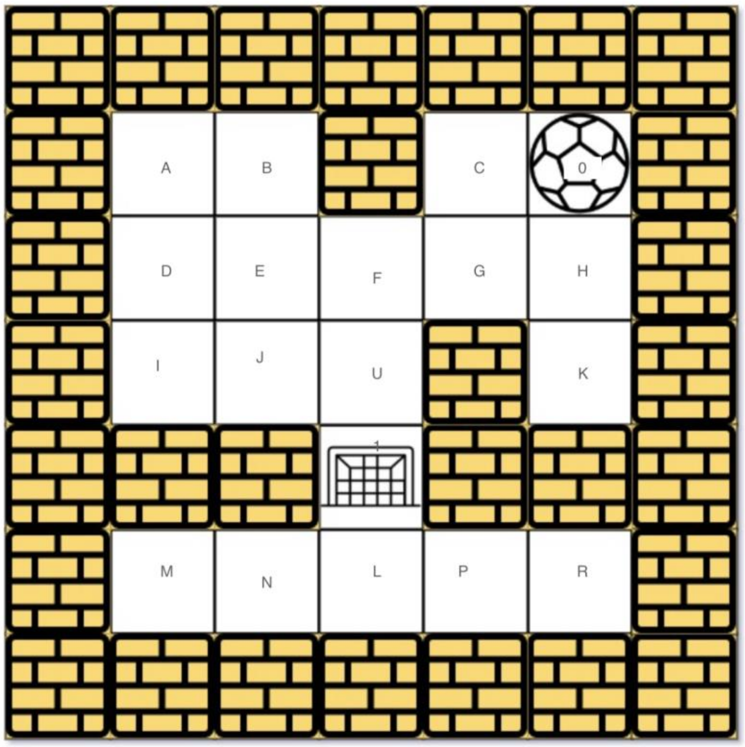
Project: Question40: “490. The Maze” - LC – Breadth-First Traversal



Visited : 0 0 Queue :	Visited : 0 C. H. K G 1 1 1 1. 1 Queue : G 1. Remove G from the queue 2. Print 0 C H K G	Visited : 0 C. H K G D A I B 1 1 1 1 1 1 1 1. 1 Queue : B 1. Remove I from the queue 2. Print 0 C H. K G D A I
Visited : 0 1 Queue : 0 1. Add 0 to the queue 2. Mark 0 as visited	Visited : 0 C. H. K G 1 1 1 1. 1 Queue : 1. Remove G from the queue 2. Print 0 C K G	Visited : 0 C H. K G D A I B U 1 1 1 1 1 1 1 1 1. 1 Queue : B U 1. Add U to the queue 2. Mark U as visited
Visited : 0 1 Queue : 1. Remove 0 from the queue 2. Print 0	Visited : 0 C. H. K G D 1 1 1 1 1. 1 Queue : D 1. Add D to the queue 2. Mark D as visited	Visited : 0 C H. K G D A I B U 1 1 1 1 1 1 1 1 1 Queue : U 1. Remove I from the queue 2. Print 0 C H K G D A I B
Visited : 0 C. H	Visited : 0 C. H. K G D	Visited : 0 C H. K G D A I B U

1 1 1 Queue : C H 1. Add C and H to the queue 2. Print 0 C	1 1 1 1 1. 1 Queue : 1. Remove D from the queue 2. Print 0 C H. K G D	1 1 1 1 1 1 1 1 1 1 Queue : 1. Remove U from the queue 2. Print 0 C H K G D A I B U
Visited : 0 C. H K G 1 1 1 1 1 Queue : H K G 1. Add K, G to the queue 2. Mark K, G as visited	Visited : 0 C. H K G D A I 1 1 1 1 1 1 1. 1 Queue : A I 1. Add A I to the queue 2. Mark A, I as visited	Visited : 0 C H K G D A I B U P 1 1 1 1 1 1 1 1 1 1 1 Queue : P 1. Add P to the queue 2. Mark P as visited
Visited : 0 C. H K G 1 1 1 1 1 Queue : H K G 3. Remove H from the queue 4. Print 0 C. H	Visited : 0 C. H. K G D A I 1 1 1 1 1 1 1. 1 Queue : I 1. Remove A from the queue 2. Print 0 C H. K G D A	Visited : 0 C H K G D A I B U P 1 1 1 1 1 1 1 1 1 1. 1 Queue : 1. Remove P from the queue 2. Print 0 C H K G D A I B U P
Visited : 0 C. H K G 1 1 1 1 1 Queue : K G 5. Remove K from the queue 6. Print 0 C. H K	Visited : 0 C. H. K G D A I B 1 1 1 1 1 1 1 1. 1 Queue : I B 1. Add B to the queue 2. Mark B as visited	Visited : 0 C H K G D A I B U P R 1 1 1 1 1 1 1 1 1 1 1 1 Queue : R 1. Add R to the queue 2. Mark R as visited
		Visited : 0 C H K G D A I B U P 1 1 1 1 1 1 1 1 1 1. 1 Queue : 3. Remove R from the queue 4. Print 0 C H K G D A I B U P R




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

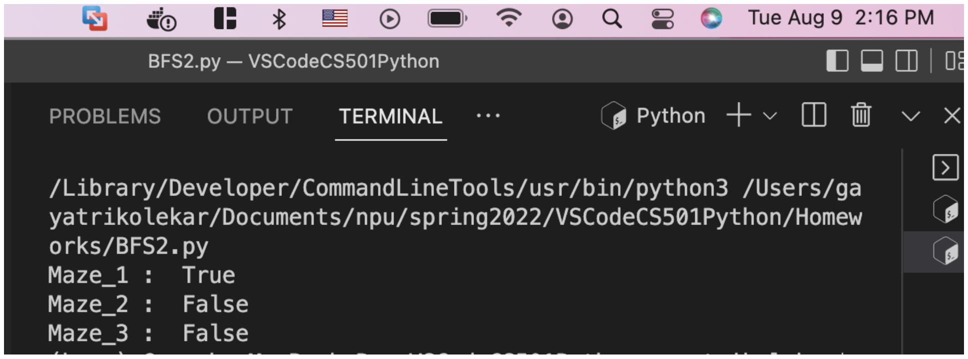
Visited : 0 0 Queue :	Visited : 0 C. H. K G 1 1 1 1. 1 Queue : G 3. Remove G from the queue 4. Print 0 C H K G	Visited : 0 C. H K G D A I B 1 1 1 1 1 1 1 1. 1 Queue : B 3. Remove I from the queue 4. Print 0 C H. K G D A I
Visited : 0 1 Queue : 0 3. Add 0 to the queue 4. Mark 0 as visited	Visited : 0 C. H. K G 1 1 1 1. 1 Queue : 3. Remove G from the queue 4. Print 0 C H K G	Visited : 0 C H. K G D A I B U 1 1 1 1 1 1 1 1 1 1 Queue : B U 3. Add U to the queue 4. Mark U as visited
Visited : 0 1 Queue : 7. Remove 0 from the queue 8. Print 0	Visited : 0 C. H. K G D 1 1 1 1 1. 1 Queue : D 3. Add D to the queue 4. Mark D as visited	Visited : 0 C H. K G D A I B U 1 1 1 1 1 1 1 1 1 Queue : U 3. Remove I from the queue 4. Print 0 C H K G D A I B
Visited : 0 C. H 1 1 1 Queue : C H 3. Add C and H to the queue 4. Print 0 C H	Visited : 0 C. H. K G D 1 1 1 1 1. 1 Queue : 3. Remove D from the queue 4. Print 0 C H. K G D	Visited : 0 C H. K G D A I B U 1 1 1 1 1 1 1 1 1 1 Queue : 3. Remove U from the queue 4. Print 0 C H K G D A I B U
Visited : 0 C. H K G 1 1 1 1 1 Queue : H K G 3. Add K, G to the queue 4. Mark K, G as visited	Visited : 0 C. H K G D A I 1 1 1 1 1 1 1. 1 Queue : A I 3. Add A I to the queue 4. Mark A, I as visited	Visited : 0 C H K G D A I B U 1 1 1 1 1 1 1 1 1 1 1 1 1 Queue : 1 3. Add 1 to the queue 4. Mark 1 as visited
Visited : 0 C. H K G 1 1 1 1 1 Queue : H K G 9. Remove H from the queue 10. Print 0 C. H	Visited : 0 C. H. K G D A I 1 1 1 1 1 1 1. 1 Queue : I 3. Remove A from the queue 4. Print 0 C H. K G D A	Visited : 0 C H K G D A I B U 1 1 1 1 1 1 1 1 1 1 1 1 1 1 Queue : 5. Remove 1 from the queue 6. Print 0 C H K G D A I B U 1
Visited : 0 C. H K G 1 1 1 1 1	Visited : 0 C. H. K G D A I B 1 1 1 1 1 1 1 1. 1	

Queue : K G 11. Remove K from the queue 12. Print 0 C. H K	Queue : I B 3. Add B to the queue 4. Mark B as visited	

Python Program Code :

```
import collections

from typing import List



def hasPath(maze: List[List[int]], start: List[int], destination: List[int]) -> bool:

    row, col = len(maze),len(maze[0])

    queue = collections.deque([(start[0],start[1])])

    visited = set()

    dirs = [(-1,0),(0,-1),(1,0),(0,1)]

    def neighbors(x,y):

        temp=[]

        used = set()

        used.add((x,y))

        for dx, dy in dirs:

            nx,ny = x,y

            while 0 <= nx+dx < row and 0 <= ny+dy < col and maze[nx+dx][ny+dy] == 0:

                nx+=dx

                ny+=dy

            if (nx,ny) not in used:

                temp.append((nx, ny))

        return temp
    while queue:
        cell = queue.popleft()
        if cell in visited: continue
        if cell == (destination[0], destination[1]): return True
        visited.add(cell)
        for neighbor in neighbors(cell[0],cell[1]):
            queue.append(neighbor)
    return False


maze_1 = [[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_1 = [0, 4]
destination_1 = [4, 4]

maze_2 = [[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_2 = [0, 4]
destination_2 = [3, 2]

maze_3 = [[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_3 = [4, 3]
destination_3 = [0, 1]

print("Maze_1 : ",hasPath(maze_1, start_1, destination_1))
print("Maze_2 : ",hasPath(maze_2, start_2, destination_2))
print("Maze_3 : ",hasPath(maze_3, start_3, destination_3))
```

Python Program Code Test Output :



```
/Library/Developer/CommandLineTools/usr/bin/python3 /Users/ga
yatrikolekar/Documents/npu/spring2022/VSCodeCS501Python/Homew
orks/BFS2.py
Maze_1 :  True
Maze_2 :  False
Maze_3 :  False
```

Google Slides:

https://docs.google.com/presentation/d/1TaaeuW-dZuGOJcp0TAsIkaOIQomVxHGOE_gqWP0oxZw/edit?usp=sharing