Problem 5: Rat in a Maze

Consider a rat placed at **(0, 0)** in a square matrix of order **N * N**. It has to reach the destination at **(N - 1, N - 1)**. Find all possible paths that the rat can take to reach from source to destination. The directions in which the rat can move are **'U'(up)**, **'D'(down)**, **'L' (left)**, **'R' (right)**. Value 0 at a cell in the matrix represents that it is blocked and the rat cannot move to it while value 1 at a cell in the matrix represents that rat can travel through it.

```
def findPaths(maze, row, col, path, paths):
    N = len(maze)
    if row == N - 1 and col == N - 1:
        paths.append(path)
        return

    if (
        row < 0
        or col < 0
        or row >= N
        or col >= N
        or maze[row][col] == 0
    ):
        return

    maze[row][col] = 0

    findPaths(maze, row - 1, col, path + "U", paths)
    findPaths(maze, row + 1, col, path + "D", paths)
    findPaths(maze, row, col - 1, path + "L", paths)
    findPaths(maze, row, col + 1, path + "R", paths)
    maze[row][col] = 1


def findMazePaths(N, m):
```

```
    paths = []
    findPaths(m, 0, 0, "", paths)
    paths.sort()
    return paths


N = 4
m = [
    [1, 0, 0, 0],
    [1, 1, 0, 1],
    [1, 1, 0, 0],
    [0, 1, 1, 1],
]
result = findMazePaths(N, m)
print(" ".join(result))
```