Our Solution(s) Run Code

Solution 1

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(wh) time | O(wh) space
def riverSizes(matrix):
    sizes = []
    visited = [[False for value in row] for row in matrix]
    for i in range(len(matrix)):
        for j in range(len(matrix[i])):
            if visited[i][j]:
                continue
            traverseNode(i, j, matrix, visited, sizes)
    return sizes


def traverseNode(i, j, matrix, visited, sizes):
    currentRiverSize = 0
    nodesToExplore = [[i, j]]
    while len(nodesToExplore):
        currentNode = nodesToExplore.pop()
        i = currentNode[0]
        j = currentNode[1]
        if visited[i][j]:
            continue
        visited[i][j] = True
        if matrix[i][j] == 0:
            continue
        currentRiverSize += 1
        unvisitedNeighbors = getUnvisitedNeighbors(i, j, matrix, visited)
        for neighbor in unvisitedNeighbors:
            nodesToExplore.append(neighbor)
    if currentRiverSize > 0:
        sizes.append(currentRiverSize)


def getUnvisitedNeighbors(i, j, matrix, visited):
    unvisitedNeighbors = []
    if i > 0 and not visited[i - 1][j]:
        unvisitedNeighbors.append([i - 1, j])
    if i < len(matrix) - 1 and not visited[i + 1][j]:
        unvisitedNeighbors.append([i + 1, j])
    if j > 0 and not visited[i][j - 1]:
        unvisitedNeighbors.append([i, j - 1])
    if j < len(matrix[0]) - 1 and not visited[i][j + 1]:
        unvisitedNeighbors.append([i, j + 1])
    return unvisitedNeighbors

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```python
def riverSizes(matrix):
    # Write your code here.
    pass

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.