

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 // O(wh) time | O(wh) space
6 func RiverSizes(matrix [][]int) []int {
7     sizes := []int{}
8     visited := make([][]bool, len(matrix))
9     for i := range visited {
10         visited[i] = make([]bool, len(matrix[i]))
11     }
12     for i := range matrix {
13         for j := range matrix[i] {
14             if visited[i][j] {
15                 continue
16             }
17             sizes = traverseNode(i, j, matrix, visited, sizes)
18         }
19     }
20     return sizes
21 }
22
23 func traverseNode(i, j int, matrix [][]int, visited [][]bool, sizes []int) []int {
24     currentRiverSize := 0
25     nodesToExplore := [][]int{{i, j}}
26     for len(nodesToExplore) > 0 {
27         currentNode := nodesToExplore[0]
28         nodesToExplore = nodesToExplore[1:]
29         i, j := currentNode[0], currentNode[1]
30         if visited[i][j] {
31             continue
32         }
33         visited[i][j] = true
34         if matrix[i][j] == 0 {
35             continue
36         }
37         currentRiverSize += 1
38         unvisitedNeighbors := getUnvisitedNeighbors(i, j, matrix, visited)
39         for _, neighbor := range unvisitedNeighbors {
40             nodesToExplore = append(nodesToExplore, neighbor)
41         }
42     }
43     if currentRiverSize > 0 {
44         sizes = append(sizes, currentRiverSize)
45     }
46     return sizes
47 }
48
49 func getUnvisitedNeighbors(i, j int, matrix [][]int, visited [][]bool) [][]int {
50     unvisitedNeighbors := [][]int{}
51     if i > 0 && !visited[i-1][j] {
52         unvisitedNeighbors = append(unvisitedNeighbors, []int{i - 1, j})
53     }
54     if i < len(matrix)-1 && !visited[i+1][j] {
55         unvisitedNeighbors = append(unvisitedNeighbors, []int{i + 1, j})
56     }
57     if j > 0 && !visited[i][j-1] {
58         unvisitedNeighbors = append(unvisitedNeighbors, []int{i, j - 1})
59     }
60     if j < len(matrix[0])-1 && !visited[i][j+1] {
61         unvisitedNeighbors = append(unvisitedNeighbors, []int{i, j + 1})
62     }
63     return unvisitedNeighbors
64 }
65
```

Solution 1 Solution 2 Solution 3

```
1 package main
2
3 func RiverSizes(matrix [][]int) []int {
4     // Write your code here.
5     return nil
6 }
7
```

Run or submit code when you're ready.