Prompt | Scratchpad | Our Solution(s) | Video Explanation | Run Code

Solution 1

```
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

using System.Collections.Generic;

public class Program {
  // O(wh) time | O(wh) space
  public static List<int> RiverSizes(int[,] matrix) {
    List<int> sizes = new List<int>();
    bool[,] visited = new bool[matrix.GetLength(0),matrix.GetLength(1)];
    for (int i = 0; i < matrix.GetLength(0); i++) {
      for (int j = 0; j < matrix.GetLength(1); j++) {
        if (visited[i,j]) {
          continue;
        }
        traverseNode(i, j, matrix, visited, sizes);
      }
    }
    return sizes;
  }

  public static void traverseNode(int i, int j, int[,] matrix, bool[,] visited,
    List<int> sizes) {
    int currentRiverSize = 0;
    List<int[]> nodesToExplore = new List<int[]>();
    nodesToExplore.Add(new int[] {i, j});
    while (nodesToExplore.Count != 0) {
      int[] currentNode = nodesToExplore[nodesToExplore.Count - 1];
      nodesToExplore.RemoveAt(nodesToExplore.Count - 1);
      i = currentNode[0];
      j = currentNode[1];
      if (visited[i,j]) {
        continue;
      }
      visited[i,j] = true;
      if (matrix[i,j] == 0) {
        continue;
      }
      currentRiverSize++;
      List<int[]> unvisitedNeighbors =
        getUnvisitedNeighbors(i, j, matrix, visited);
      foreach (int[] neighbor in unvisitedNeighbors) {
        nodesToExplore.Add(neighbor);
      }
    }
    if (currentRiverSize > 0) {
      sizes.Add(currentRiverSize);
    }
  }

  public static List<int[]> getUnvisitedNeighbors(int i, int j, int[,] matrix, bo
    ] visited) {
    List<int[]> unvisitedNeighbors = new List<int[]>();
    if (i > 0 && !visited[i - 1,j]) {
      unvisitedNeighbors.Add(new int[] {i - 1, j});
    }
    if (i < matrix.GetLength(0) - 1 && !visited[i + 1,j]) {
      unvisitedNeighbors.Add(new int[] {i + 1, j});
    }
    if (j > 0 && !visited[i,j - 1]) {
      unvisitedNeighbors.Add(new int[] {i, j - 1});
    }
    if (j < matrix.GetLength(1) - 1 && !visited[i,j + 1]) {
      unvisitedNeighbors.Add(new int[] {i, j + 1});
    }
    return unvisitedNeighbors;
  }
}

```

Your Solutions | Run Code

Solution 1 | Solution 2 | Solution 3

```
using System.Collections.Generic;

public class Program {
  public static List<int> RiverSizes(int[,] matrix) {
    // Write your code here.
    return null;
  }
}

```

Custom Output | Raw Output | Submit Code

Run or submit code when you're ready.