Our Solution(s)   Run Code

Solution 1

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

import java.util.*;

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

  public static void traverseNode(
      int i, int j, int[][] matrix, boolean[][] visited, List<In
    int currentRiverSize = 0;
    List<Integer[]> nodesToExplore = new ArrayList<Integer[]>();
    nodesToExplore.add(new Integer[] {i, j});
    while (!nodesToExplore.isEmpty()) {
      Integer[] currentNode = nodesToExplore.get(nodesToExplore.
      nodesToExplore.remove(nodesToExplore.size() - 1);
      i = currentNode[0];
      j = currentNode[1];
      if (visited[i][j]) {
        continue;
      }
```

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```java
import java.util.*;

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

Our Tests   Custom Output   Submit Code

```
class ProgramTest {
    @Test
    public void TestCase01() {
        int[] input = {...};
        int[] expected = {...};
        List<Integer> output = Program.listToList(input);
        Collections.sort(output);
        Utils.assertTrue(compare(output, expected));
    }

    @Test
    public void TestCase01() {
        int[] input = {...};
        int[] expected = {...};
        List<Integer> output = Program.listToList(input);
        Collections.sort(output);
        Utils.assertTrue(compare(output, expected));
    }

    @Test
```

Run or submit code when you're ready.