Our Solution(s) Run Code

Solution 1 Solution 2

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

import java.util.*;

class Program {
  // O(n) time | O(n) space - where n is the total number of elements in the array
  public static List<Integer> spiralTraverse(int[][] array) {
    if (array.length == 0) return new ArrayList<Integer>();

    var result = new ArrayList<Integer>();
    spiralFill(array, 0, array.length - 1, 0, array[0].length - 1, result);
    return result;
  }

  public static void spiralFill(
      int[][] array,
      int startRow,
      int endRow,
      int startCol,
      int endCol,
      ArrayList<Integer> result) {
    if (startRow > endRow || startCol > endCol) {
      return;
    }

    for (int col = startCol; col <= endCol; col++) {
      result.add(array[startRow][col]);
    }

    for (int row = startRow + 1; row <= endRow; row++) {
      result.add(array[row][endCol]);
    }

    for (int col = endCol - 1; col >= startCol; col--) {
      if (startRow == endRow) break;
      result.add(array[endRow][col]);
    }

    for (int row = endRow - 1; row >= startRow + 1; row--) {
      if (startCol == endCol) break;
      result.add(array[row][startCol]);
    }
    spiralFill(array, startRow + 1, endRow - 1, startCol + 1, endCol - 1, result);
  }
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
import java.util.*;

class Program {
  public static List<Integer> spiralTraverse(int[][] array) {
    // Write your code here.
    return new ArrayList<Integer>();
  }
}

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.