AlgoExpert Quad Layout Java 14px Sublime Monokai 00:00:

Our Solution(s) Run Code

Solution 1 Solution 2

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

import java.util.*;

class Program {
  // O(n^2) time | O(d) space - where n is the number of
  // nodes in each array, respectively, and d is the depth
  // of the BST that they represent
  public static boolean sameBsts(List<Integer> arrayOne, List<Integer
    return areSameBsts(arrayOne, arrayTwo, 0, 0, Integer.MIN_VALUE, In
  }

  public static boolean areSameBsts(
      List<Integer> arrayOne,
      List<Integer> arrayTwo,
      int rootIdxOne,
      int rootIdxTwo,
      int minVal,
      int maxVal) {
    if (rootIdxOne == -1 || rootIdxTwo == -1) return rootIdxOne == roo

    if (arrayOne.get(rootIdxOne).intValue() != arrayTwo.get(rootIdxTwo

    int leftRootIdxOne = getIdxOfFirstSmaller(arrayOne, rootIdxOne, mi
    int leftRootIdxTwo = getIdxOfFirstSmaller(arrayTwo, rootIdxTwo, mi
    int rightRootIdxOne = getIdxOfFirstBiggerOrEqual(arrayOne, rootIdx
    int rightRootIdxTwo = getIdxOfFirstBiggerOrEqual(arrayTwo, rootIdx

    int currentValue = arrayOne.get(rootIdxOne);
    boolean leftAreSame =
        areSameBsts(arrayOne, arrayTwo, leftRootIdxOne, leftRootIdxTwo
    boolean rightAreSame =
        areSameBsts(arrayOne, arrayTwo, rightRootIdxOne, rightRootIdxT
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
import java.util.*;

class Program {
  public static boolean sameBsts(List<Integer> arrayOne, List<Integer>
    // Write your code here.
    return false;
  }
}
```

Our Tests

```java
import java.util.*;

class ProgramTest {
  @Test
  public void TestCase1() {
    List<Integer> arrayOne = new ArrayList<Integer>(Arrays.asList(
    List<Integer> arrayTwo = new ArrayList<Integer>(Arrays.asList(
```

Custom Output Submit Code

Run or submit code when you're ready.