

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.*;
4
5 class Program {
6     // O(n^2) time | O(n^2) space - where n is the number of
7     // nodes in each array, respectively
8     public static boolean sameBsts(List<Integer> arrayOne, List<Integer> arrayTwo) {
9         if (arrayOne.size() != arrayTwo.size()) return false;
10
11         if (arrayOne.size() == 0 && arrayTwo.size() == 0) return true;
12
13         if (arrayOne.get(0).intValue() != arrayTwo.get(0).intValue()) return false;
14
15         List<Integer> leftOne = getSmaller(arrayOne);
16         List<Integer> leftTwo = getSmaller(arrayTwo);
17         List<Integer> rightOne = getBiggerOrEqual(arrayOne);
18         List<Integer> rightTwo = getBiggerOrEqual(arrayTwo);
19
20         return sameBsts(leftOne, leftTwo) && sameBsts(rightOne, rightTwo);
21     }
22
23     public static List<Integer> getSmaller(List<Integer> array) {
24         List<Integer> smaller = new ArrayList<Integer>();
25         for (int i = 1; i < array.size(); i++) {
26             if (array.get(i).intValue() < array.get(0).intValue()) smaller.add(array.get(i));
27         }
28         return smaller;
29     }
30
31     public static List<Integer> getBiggerOrEqual(List<Integer> array) {
32         List<Integer> biggerOrEqual = new ArrayList<Integer>();
33         for (int i = 1; i < array.size(); i++) {
34             if (array.get(i).intValue() >= array.get(0).intValue()) biggerOrEqual.add(array.get(i));
35         }
36         return biggerOrEqual;
37     }
38 }
39
```