

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 // O(n^2) time | O(n^2) space - where n is the number of
6 // nodes in each array, respectively
7 func SameBSTs(arrayOne, arrayTwo []int) bool {
8     if len(arrayOne) != len(arrayTwo) {
9         return false
10    }
11    if len(arrayOne) == 0 && len(arrayTwo) == 0 {
12        return true
13    }
14
15    if arrayOne[0] != arrayTwo[0] {
16        return false
17    }
18
19    leftOne := getSmaller(arrayOne)
20    leftTwo := getSmaller(arrayTwo)
21    rightOne := getBiggerOrEqual(arrayOne)
22    rightTwo := getBiggerOrEqual(arrayTwo)
23    return SameBSTs(leftOne, leftTwo) && SameBSTs(rightOne, rightTwo)
24 }
25
26 func getSmaller(array []int) []int {
27     smaller := []int{}
28     for i := 1; i < len(array); i++ {
29         if array[i] < array[0] {
30             smaller = append(smaller, array[i])
31         }
32     }
33     return smaller
34 }
35
36 func getBiggerOrEqual(array []int) []int {
37     biggerOrEqual := []int{}
38     for i := 1; i < len(array); i++ {
39         if array[i] >= array[0] {
40             biggerOrEqual = append(biggerOrEqual, array[i])
41         }
42     }
43     return biggerOrEqual
44 }
45
```