Solution 1 Solution 2

Run Code

Our Solution(s)

Run Code

Your Solutions

Solution 1 Solution 2 Solution 3

```
package main

func SameBSTs(arrayOne, arrayTwo []int) bool {
   // Write your code here.
   return false
}
```

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

Run or submit code when you're ready.