9

10

11 12 13

14

16 17

18

19

20

21

22 24

26

27

28

29 30 31

32 33 **Your Solutions**

Run Code

Our Solution(s) Run Code

```
Solution 1 Solution 2
    // O(n^2) time | O(n^2) space - where n is the number of
    // nodes in each array, respectively
    func sameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int]) -> Bool {
```

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
class Program {
        if arrayOne.count != arrayTwo.count {
            return false
        } else if arrayOne.count == 0, arrayTwo.count == 0 {
            return true
        if arrayOne[0] != arrayTwo[0] {
            return false
        let leftOne = getSmaller(arrayOne)
        let leftTwo = getSmaller(arrayTwo)
```

let rightOne = getBiggerOrEqual(arrayOne)

let rightTwo = getBiggerOrEqual(arrayTwo)

smaller.append(array[i])

func getSmaller(_ array: [Int]) -> [Int] {

var smaller = [Int]()

return smaller

for i in 1 ..< array.count {</pre>

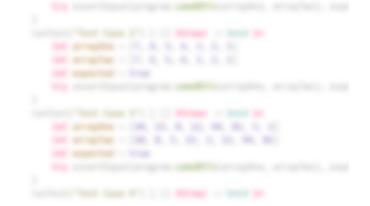
if array[i] < array[0] {</pre>

return sameBSTs(leftOne, leftTwo) && sameBSTs(rightOne, rightT

```
14рх
           Sublime
```

Solution 1 Solution 2 Solution 3

```
1 class Program {
    func sameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int]) -> Bool {
        // Write your code here.
        return false
```



Run or submit code when you're ready.