

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n^2) time | O(n^2) space - where n is the number of
5     // nodes in each array, respectively
6     func sameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int]) -> Bool {
7         if arrayOne.count != arrayTwo.count {
8             return false
9         } else if arrayOne.count == 0, arrayTwo.count == 0 {
10             return true
11         }
12
13         if arrayOne[0] != arrayTwo[0] {
14             return false
15         }
16
17         let leftOne = getSmaller(arrayOne)
18         let leftTwo = getSmaller(arrayTwo)
19         let rightOne = getBiggerOrEqual(arrayOne)
20         let rightTwo = getBiggerOrEqual(arrayTwo)
21         return sameBSTs(leftOne, leftTwo) && sameBSTs(rightOne, rightTwo)
22     }
23
24     func getSmaller(_ array: [Int]) -> [Int] {
25         var smaller = [Int]()
26         for i in 1 ..< array.count {
27             if array[i] < array[0] {
28                 smaller.append(array[i])
29             }
30         }
31         return smaller
32     }
33 }
```

Solution 1

Solution 2

Solution 3

```
1 class Program {
2     func sameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int]) -> Bool {
3         // Write your code here.
4         return false
5     }
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 class Program {
2     func sameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int]) -> Bool {
3         // Write your code here.
4         return false
5     }
6 }
```

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n^2) time | O(n^2) space - where n is the number of
5     // nodes in each array, respectively
6     func sameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int]) -> Bool {
7         if arrayOne.count != arrayTwo.count {
8             return false
9         } else if arrayOne.count == 0, arrayTwo.count == 0 {
10             return true
11         }
12
13         if arrayOne[0] != arrayTwo[0] {
14             return false
15         }
16
17         let leftOne = getSmaller(arrayOne)
18         let leftTwo = getSmaller(arrayTwo)
19         let rightOne = getBiggerOrEqual(arrayOne)
20         let rightTwo = getBiggerOrEqual(arrayTwo)
21         return sameBSTs(leftOne, leftTwo) && sameBSTs(rightOne, rightTwo)
22     }
23
24     func getSmaller(_ array: [Int]) -> [Int] {
25         var smaller = [Int]()
26         for i in 1 ..< array.count {
27             if array[i] < array[0] {
28                 smaller.append(array[i])
29             }
30         }
31         return smaller
32     }
33 }
```

Run or submit code when you're ready.