

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(d) space - where n is the number of
5     // nodes in each array, respectively, and d is the depth
6     // of the BST that they represent
7     func sameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int]) -> Bool {
8         return areSameBSTs(arrayOne, arrayTwo, 0, 0, Int.min, Int.max)
9     }
10
11     func areSameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int], _ rootIdxOne: Int, _ rootIdxTwo: Int) -> Bool {
12         if rootIdxOne == -1 || rootIdxTwo == -1 {
13             return rootIdxOne == rootIdxTwo
14         }
15
16         if arrayOne[rootIdxOne] != arrayTwo[rootIdxTwo] {
17             return false
18         }
19
20         let leftRootIdxOne = getIdxOfFirstSmaller(arrayOne, rootIdxOne)
21         let leftRootIdxTwo = getIdxOfFirstSmaller(arrayTwo, rootIdxTwo)
22         let rightRootIdxOne = getIdxOfFirstBiggerOrEqual(arrayOne, rootIdxOne)
23         let rightRootIdxTwo = getIdxOfFirstBiggerOrEqual(arrayTwo, rootIdxTwo)
24
25         let currentValue = arrayOne[rootIdxOne]
26         let leftAreSame = areSameBSTs(arrayOne, arrayTwo, leftRootIdxOne, leftRootIdxTwo)
27         let rightAreSame = areSameBSTs(arrayOne, arrayTwo, rightRootIdxOne, rightRootIdxTwo)
28
29         return leftAreSame && rightAreSame
30     }
31
32     func getIdxOfFirstSmaller(_ array: [Int], _ startingIdx: Int, _ min: Int) -> Int {
33         // Find the index of the first smaller value after the startin
```

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 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n^2) time | O(d) space - where n is the number of
5     // nodes in each array, respectively, and d is the depth
6     // of the BST that they represent
7     func sameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int]) -> Bool {
8         return areSameBSTs(arrayOne, arrayTwo, 0, 0, Int.min, Int.max)
9     }
10
11     func areSameBSTs(_ arrayOne: [Int], _ arrayTwo: [Int], _ rootIdxOne: Int, _ rootIdxTwo: Int) -> Bool {
12         if rootIdxOne == -1 || rootIdxTwo == -1 {
13             return rootIdxOne == rootIdxTwo
14         }
15
16         if arrayOne[rootIdxOne] != arrayTwo[rootIdxTwo] {
17             return false
18         }
19
20         let leftRootIdxOne = getIdxOfFirstSmaller(arrayOne, rootIdxOne)
21         let leftRootIdxTwo = getIdxOfFirstSmaller(arrayTwo, rootIdxTwo)
22         let rightRootIdxOne = getIdxOfFirstBiggerOrEqual(arrayOne, rootIdxOne)
23         let rightRootIdxTwo = getIdxOfFirstBiggerOrEqual(arrayTwo, rootIdxTwo)
24
25         let currentValue = arrayOne[rootIdxOne]
26         let leftAreSame = areSameBSTs(arrayOne, arrayTwo, leftRootIdxOne, leftRootIdxTwo)
27         let rightAreSame = areSameBSTs(arrayOne, arrayTwo, rightRootIdxOne, rightRootIdxTwo)
28
29         return leftAreSame && rightAreSame
30     }
31
32     func getIdxOfFirstSmaller(_ array: [Int], _ startingIdx: Int, _ min: Int) -> Int {
33         // Find the index of the first smaller value after the startin
```

```

18         assert isinstance(average, np.ndarray)
19     )
20     assert isinstance(average, np.ndarray)
21     assert average.shape == (10, 10)
22     assert average.dtype == np.float64
23     assert average.min() > 0
24     assert average.max() < 1
25     assert average.sum() > 0
26     assert average.sum() < 100
27     assert average.sum() > 0
28     assert average.sum() < 100
29     assert average.sum() > 0
30     assert average.sum() < 100
31     assert average.sum() > 0
32     assert average.sum() < 100
33     assert average.sum() > 0
34     assert average.sum() < 100
35     assert average.sum() > 0
36     assert average.sum() < 100
37     assert average.sum() > 0
38     assert average.sum() < 100
39     assert average.sum() > 0
40     assert average.sum() < 100
41     assert average.sum() > 0
42     assert average.sum() < 100
43     assert average.sum() > 0
44     assert average.sum() < 100
45     assert average.sum() > 0
46     assert average.sum() < 100
47     assert average.sum() > 0
48     assert average.sum() < 100
49     assert average.sum() > 0
50     assert average.sum() < 100
51     assert average.sum() > 0
52     assert average.sum() < 100
53     assert average.sum() > 0
54     assert average.sum() < 100
55     assert average.sum() > 0
56     assert average.sum() < 100
57     assert average.sum() > 0
58     assert average.sum() < 100
59     assert average.sum() > 0
60     assert average.sum() < 100
61     assert average.sum() > 0
62     assert average.sum() < 100
63     assert average.sum() > 0
64     assert average.sum() < 100
65     assert average.sum() > 0
66     assert average.sum() < 100
67     assert average.sum() > 0
68     assert average.sum() < 100
69     assert average.sum() > 0
70     assert average.sum() < 100
71     assert average.sum() > 0
72     assert average.sum() < 100
73     assert average.sum() > 0
74     assert average.sum() < 100
75     assert average.sum() > 0
76     assert average.sum() < 100
77     assert average.sum() > 0
78     assert average.sum() < 100
79     assert average.sum() > 0
80     assert average.sum() < 100
81     assert average.sum() > 0
82     assert average.sum() < 100
83     assert average.sum() > 0
84     assert average.sum() < 100
85     assert average.sum() > 0
86     assert average.sum() < 100
87     assert average.sum() > 0
88     assert average.sum() < 100
89     assert average.sum() > 0
90     assert average.sum() < 100
91     assert average.sum() > 0
92     assert average.sum() < 100
93     assert average.sum() > 0
94     assert average.sum() < 100
95     assert average.sum() > 0
96     assert average.sum() < 100
97     assert average.sum() > 0
98     assert average.sum() < 100
99     assert average.sum() > 0
100    assert average.sum() < 100

```

Run or submit code when you're ready.