Our Solution(s)

Run Code

Your Solutions

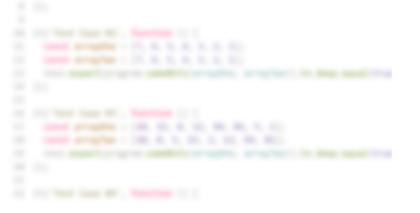
Run Code

```
Solution 1 Solution 2
 1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
 3 // O(n^2) time | O(d) space - where n is the number of
 4 // nodes in each array, respectively, and d is the depth
 5 // of the BST that they represent
 6 function sameBsts(arrayOne, arrayTwo) {
     return areSameBsts(arrayOne, arrayTwo, 0, 0, -Infinity, Infinity);
8
9
10 function areSameBsts(arrayOne, arrayTwo, rootIdxOne, rootIdxTwo, minV
11
     if (rootIdxOne === -1 || rootIdxTwo === -1) return rootIdxOne === rule
12
     if (arrayOne[rootIdxOne] !== arrayTwo[rootIdxTwo]) return false;
14
15
     const leftRootIdxOne = getIdxOfFirstSmaller(arrayOne, rootIdxOne, mi
     const leftRootIdxTwo = getIdxOfFirstSmaller(arrayTwo, rootIdxTwo, mi
16
17
     const rightRootIdxOne = getIdxOfFirstBiggerOrEqual(arrayOne, rootIdx
     const rightRootIdxTwo = getIdxOfFirstBiggerOrEqual(arrayTwo, rootIdx
18
19
20
     const currentValue = arrayOne[rootIdxOne];
21
     const leftAreSame = areSameBsts(arrayOne, arrayTwo, leftRootIdxOne,
     const rightAreSame = areSameBsts(arrayOne, arrayTwo, rightRootIdxOne
24
     return leftAreSame && rightAreSame;
25 }
26
27
   function getIdxOfFirstSmaller(array, startingIdx, minVal) {
     // Find the index of the first smaller value after the startingIdx.
28
29
     // Make sure that this value is greater than or equal to the minVal,
     \ensuremath{//} which is the value of the previous parent node in the BST. If it
30
31
     // isn't, then that value is located in the left subtree of the
     // previous parent node.
     for (let i = startingIdx + 1; i < array.length; i++) {</pre>
```

```
function sameBsts(arrayOne, arrayTwo) {
   // Write your code here.
}

// Do not edit the line below.
exports.sameBsts = sameBsts;
```

 Our Tests
 Custom Output
 Submit Code



Run or submit code when you're ready.