

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
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 def sameBsts(arrayOne, arrayTwo):
7     return areSameBsts(arrayOne, arrayTwo, 0, 0, float("-inf"), float("inf"))
8
9
10 def areSameBsts(arrayOne, arrayTwo, rootIdxOne, rootIdxTwo, minVal, maxVal):
11     if rootIdxOne == -1 or rootIdxTwo == -1:
12         return rootIdxOne == rootIdxTwo
13
14     if arrayOne[rootIdxOne] != arrayTwo[rootIdxTwo]:
15         return False
16
17     leftRootIdxOne = getIdxOfFirstSmaller(arrayOne, rootIdxOne, minVal)
18     leftRootIdxTwo = getIdxOfFirstSmaller(arrayTwo, rootIdxTwo, minVal)
19     rightRootIdxOne = getIdxOfFirstBiggerOrEqual(arrayOne, rootIdxOne, maxVal)
20     rightRootIdxTwo = getIdxOfFirstBiggerOrEqual(arrayTwo, rootIdxTwo, maxVal)
21
22     currentValue = arrayOne[rootIdxOne]
23     leftAreSame = areSameBsts(arrayOne, arrayTwo, leftRootIdxOne, leftRootIdxTwo, minVal, currentValue)
24     rightAreSame = areSameBsts(arrayOne, arrayTwo, rightRootIdxOne, rightRootIdxTwo, currentValue, maxVal)
25
26     return leftAreSame and rightAreSame
27
28
29 def getIdxOfFirstSmaller(array, startingIdx, minVal):
30     # Find the index of the first smaller value after the startingIdx.
31     # Make sure that this value is greater than or equal to the minVal,
32     # which is the value of the previous parent node in the BST. If it
33     # isn't, then that value is located in the left subtree of the
34     # previous parent node.
35     for i in range(startingIdx + 1, len(array)):
36         if array[i] < array[startingIdx] and array[i] >= minVal:
37             return i
38     return -1
39
40
41 def getIdxOfFirstBiggerOrEqual(array, startingIdx, maxVal):
42     # Find the index of the first bigger/equal value after the startingIdx.
43     # Make sure that this value is smaller than maxVal, which is the value
44     # of the previous parent node in the BST. If it isn't, then that value
45     # is located in the right subtree of the previous parent node.
46     for i in range(startingIdx + 1, len(array)):
47         if array[i] >= array[startingIdx] and array[i] < maxVal:
48             return i
49     return -1
50
```