Our Solution(s)   Run Code

Solution 1   Solution 2

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(n^2) time | O(d) space - where n is the number of
# nodes in each array, respectively, and d is the depth
# of the BST that they represent
def sameBsts(arrayOne, arrayTwo):
    return areSameBsts(arrayOne, arrayTwo, 0, 0, float("-inf"), float


def areSameBsts(arrayOne, arrayTwo, rootIdxOne, rootIdxTwo, minVal, m
    if rootIdxOne == -1 or rootIdxTwo == -1:
        return rootIdxOne == rootIdxTwo

    if arrayOne[rootIdxOne] != arrayTwo[rootIdxTwo]:
        return False

    leftRootIdxOne = getIdxOfFirstSmaller(arrayOne, rootIdxOne, minVal
    leftRootIdxTwo = getIdxOfFirstSmaller(arrayTwo, rootIdxTwo, minVal
    rightRootIdxOne = getIdxOfFirstBiggerOrEqual(arrayOne, rootIdxOne,
    rightRootIdxTwo = getIdxOfFirstBiggerOrEqual(arrayTwo, rootIdxTwo,

    currentValue = arrayOne[rootIdxOne]
    leftAreSame = areSameBsts(arrayOne, arrayTwo, leftRootIdxOne, left
    rightAreSame = areSameBsts(arrayOne, arrayTwo, rightRootIdxOne, ri

    return leftAreSame and rightAreSame


def getIdxOfFirstSmaller(array, startingIdx, minVal):
    # Find the index of the first smaller value after the startingIdx.
    # Make sure that this value is greater than or equal to the minVal
    # which is the value of the previous parent node in the BST. If it
    # isn't, then that value is located in the left subtree of the
```

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```python
def sameBsts(arrayOne, arrayTwo):
    # Write your code here.
    pass
```

Our Tests

Custom Output   Submit Code

```
         arrayOne = [...]
         self.assertEqual(program.sortArray(arrayOne, arrayTwo), True)

    def test_case_2(self):
         arrayOne = [...]
         arrayTwo = [...]
         self.assertEqual(program.sortArray(arrayOne, arrayTwo), True)

    def test_case_3(self):
         arrayOne = [...]
         arrayTwo = [...]
         self.assertEqual(program.sortArray(arrayOne, arrayTwo), True)

    def test_case_4(self):
         arrayOne = [...]
```