```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

#include <vector>

using namespace std;

vector<int> getSmaller(vector<int>);
vector<int> getBiggerOrEqual(vector<int> array);

// O(n^2) time | O(n^2) space - where n is the number of
// nodes in each array, respectively
bool sameBsts(vector<int> arrayOne, vector<int> arrayTwo) {
  if (arrayOne.size() != arrayTwo.size())
    return false;

  if (arrayOne.size() == 0 && arrayTwo.size() == 0)
    return true;

  if (arrayOne[0] != arrayTwo[0])
    return false;

  vector<int> leftOne = getSmaller(arrayOne);
  vector<int> leftTwo = getSmaller(arrayTwo);
  vector<int> rightOne = getBiggerOrEqual(arrayOne);
  vector<int> rightTwo = getBiggerOrEqual(arrayTwo);

  return sameBsts(leftOne, leftTwo) && sameBsts(rightOne, rightTwo);
}

vector<int> getSmaller(vector<int> array) {
  vector<int> smaller = {};
  for (int i = 1; i < array.size(); i++) {
    if (array[i] < array[0])
      smaller.push_back(array[i]);
  }
  return smaller;
}

vector<int> getBiggerOrEqual(vector<int> array) {
  vector<int> biggerOrEqual = {};
  for (int i = 1; i < array.size(); i++) {
    if (array[i] >= array[0])
      biggerOrEqual.push_back(array[i]);
  }
  return biggerOrEqual;
}
```