

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4
5 using namespace std;
6
7 bool areSameBsts(vector<int> arrayOne, vector<int> arrayTwo, int rootIdxOne,
8                 int rootIdxTwo, int minVal, int maxVal);
9 int getIdxOffFirstSmaller(vector<int> array, int startingIdx, int minVal, int maxVal);
10 int getIdxOffFirstBiggerOrEqual(vector<int> array, int startingIdx, int minVal, int maxVal);
11
12 // O(n^2) time | O(d) space - where n is the number of
13 // nodes in each array, respectively, and d is the depth
14 // of the BST that they represent
15 bool sameBsts(vector<int> arrayOne, vector<int> arrayTwo) {
16     return areSameBsts(arrayOne, arrayTwo, 0, 0, INT_MIN, INT_MAX);
17 }
18
19 bool areSameBsts(vector<int> arrayOne, vector<int> arrayTwo, int rootIdxOne,
20                 int rootIdxTwo, int minVal, int maxVal) {
21     if (rootIdxOne == -1 || rootIdxTwo == -1)
22         return rootIdxOne == rootIdxTwo;
23
24     if (arrayOne[rootIdxOne] != arrayTwo[rootIdxTwo])
25         return false;
26
27     int leftRootIdxOne = getIdxOffFirstSmaller(arrayOne, rootIdxOne, minVal, arrayOne[rootIdxOne]);
28     int leftRootIdxTwo = getIdxOffFirstSmaller(arrayTwo, rootIdxTwo, minVal, arrayTwo[rootIdxTwo]);
29     int rightRootIdxOne =
30         getIdxOffFirstBiggerOrEqual(arrayOne, rootIdxOne, arrayOne[rootIdxOne], maxVal);
31     int rightRootIdxTwo =
32         getIdxOffFirstBiggerOrEqual(arrayTwo, rootIdxTwo, arrayTwo[rootIdxTwo], maxVal);
33
34     return areSameBsts(arrayOne, arrayTwo, leftRootIdxOne, leftRootIdxTwo, minVal, arrayOne[rootIdxOne]) &&
35         areSameBsts(arrayOne, arrayTwo, rightRootIdxOne, rightRootIdxTwo, arrayOne[rootIdxOne], maxVal);
36 }
```

Solution 1

Solution 2

Solution 3

```
1 #include <vector>
2
3 using namespace std;
4
5 bool sameBsts(vector<int> arrayOne, vector<int> arrayTwo) {
6     // Write your code here.
7     return false;
8 }
9
```

Our Tests

Custom Output

Submit Code

1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.

2

3 #include <vector>

4

5 using namespace std;

6

7 bool areSameBsts(vector<int> arrayOne, vector<int> arrayTwo, int rootIdxOne, int rootIdxTwo, int minVal, int maxVal);

8 int getIdxOffFirstSmaller(vector<int> array, int startingIdx, int minVal, int maxVal);

9 int getIdxOffFirstBiggerOrEqual(vector<int> array, int startingIdx, int minVal, int maxVal);

10

11

12 // 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

13

14 bool sameBsts(vector<int> arrayOne, vector<int> arrayTwo) {

15 return areSameBsts(arrayOne, arrayTwo, 0, 0, INT\_MIN, INT\_MAX);

16 }

17

18 bool areSameBsts(vector<int> arrayOne, vector<int> arrayTwo, int rootIdxOne, int rootIdxTwo, int minVal, int maxVal) {

19 if (rootIdxOne == -1 || rootIdxTwo == -1) return rootIdxOne == rootIdxTwo;

20

21 if (arrayOne[rootIdxOne] != arrayTwo[rootIdxTwo]) return false;

22

23 int leftRootIdxOne = getIdxOffFirstSmaller(arrayOne, rootIdxOne, minVal, arrayOne[rootIdxOne]);

24 int leftRootIdxTwo = getIdxOffFirstSmaller(arrayTwo, rootIdxTwo, minVal, arrayTwo[rootIdxTwo]);

25 int rightRootIdxOne = getIdxOffFirstBiggerOrEqual(arrayOne, rootIdxOne, arrayOne[rootIdxOne], maxVal);

26 int rightRootIdxTwo = getIdxOffFirstBiggerOrEqual(arrayTwo, rootIdxTwo, arrayTwo[rootIdxTwo], maxVal);

27

28 return areSameBsts(arrayOne, arrayTwo, leftRootIdxOne, leftRootIdxTwo, minVal, arrayOne[rootIdxOne]) &&

29 areSameBsts(arrayOne, arrayTwo, rightRootIdxOne, rightRootIdxTwo, arrayOne[rootIdxOne], maxVal);

30 }

1 #include <vector>

2

3 using namespace std;

4

5 bool sameBsts(vector<int> arrayOne, vector<int> arrayTwo) {

6 // Write your code here.

7 return false;

8 }

9

Run or submit code when you're ready.