

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1	Solution 2
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 import java.util.*; 4 5 class Program { 6 // O(n^2) time O(d) space - where n is the number of 7 // nodes in each array, respectively, and d is the depth 8 // of the BST that they represent 9 public static boolean sameBsts(List<Integer> arrayOne, List<Integer> arrayTwo) { 10 return areSameBsts(arrayOne, arrayTwo, 0, 0, Integer.MIN_VALUE, Integer.MAX_VALUE); 11 } 12 13 public static boolean areSameBsts(List<Integer> arrayOne, List<Integer> arrayTwo, int rootIdxOne, int rootIdxTwo, int minVal, int maxVal) { 14 if (rootIdxOne == -1 rootIdxTwo == -1) return rootIdxOne == rootIdxTwo; 15 16 if (arrayOne.get(rootIdxOne).intValue() != arrayTwo.get(rootIdxTwo).intValue()) return false; 17 18 int leftRootIdxOne = getIdxOfFirstSmaller(arrayOne, rootIdxOne, minVal); 19 int leftRootIdxTwo = getIdxOfFirstSmaller(arrayTwo, rootIdxTwo, minVal); 20 int rightRootIdxOne = getIdxOfFirstBiggerOrEqual(arrayOne, rootIdxOne, arrayOne.get(rootIdxOne).intValue()); 21 int rightRootIdxTwo = getIdxOfFirstBiggerOrEqual(arrayTwo, rootIdxTwo, arrayTwo.get(rootIdxTwo).intValue()); 22 23 boolean leftAreSame = areSameBsts(arrayOne, arrayTwo, leftRootIdxOne, leftRootIdxTwo, minVal, arrayOne.get(rootIdxOne).intValue()); 24 boolean rightAreSame = areSameBsts(arrayOne, arrayTwo, rightRootIdxOne, rightRootIdxTwo, arrayOne.get(rootIdxOne).intValue(), maxVal); 25 26 return leftAreSame && rightAreSame; 27 } 28 29 private static int getIdxOfFirstSmaller(List<Integer> array, int startIdx, int val) { 30 for (int i = startIdx; i < array.size(); i++) { 31 if (array.get(i).intValue() < val) return i; 32 } 33 return -1; 34 } 35 36 private static int getIdxOfFirstBiggerOrEqual(List<Integer> array, int startIdx, int val) { 37 for (int i = startIdx; i < array.size(); i++) { 38 if (array.get(i).intValue() >= val) return i; 39 } 40 return -1; 41 } 42 }</pre>	<pre>1 import java.util.*; 2 3 class Program { 4 public static boolean sameBsts(List<Integer> arrayOne, List<Integer> arrayTwo) { 5 // Write your code here. 6 return false; 7 } 8 } 9</pre>

Our Tests

Custom Output

Submit Code

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

```

10 while count < 10:
11     print(count)
12     count += 1
13
14 # Print out the number of times you looped over the range object,
15 # using len()
16
17 # Print out variable count, which has the value 10
18
19 # Print out range object, which has a default value of 10
20
21 # Print out variable count, which has the value 10
22
23 # Print out range object, which has a default value of 10
24
25 # Print out variable count, which has the value 10
26
27 # Print out range object, which has a default value of 10
28
29 # Print out variable count, which has the value 10
30
31 # Print out range object, which has a default value of 10
32
33 # Print out variable count, which has the value 10
34
35 # Print out range object, which has a default value of 10
36
37 # Print out variable count, which has the value 10
38
39 # Print out range object, which has a default value of 10
40
41 # Print out variable count, which has the value 10
42
43 # Print out range object, which has a default value of 10
44
45 # Print out variable count, which has the value 10
46
47 # Print out range object, which has a default value of 10
48
49 # Print out variable count, which has the value 10
50
51 # Print out range object, which has a default value of 10
52
53 # Print out variable count, which has the value 10
54
55 # Print out range object, which has a default value of 10
56
57 # Print out variable count, which has the value 10
58
59 # Print out range object, which has a default value of 10
60
61 # Print out variable count, which has the value 10
62
63 # Print out range object, which has a default value of 10
64
65 # Print out variable count, which has the value 10
66
67 # Print out range object, which has a default value of 10
68
69 # Print out variable count, which has the value 10
70
71 # Print out range object, which has a default value of 10
72
73 # Print out variable count, which has the value 10
74
75 # Print out range object, which has a default value of 10
76
77 # Print out variable count, which has the value 10
78
79 # Print out range object, which has a default value of 10
80
81 # Print out variable count, which has the value 10
82
83 # Print out range object, which has a default value of 10
84
85 # Print out variable count, which has the value 10
86
87 # Print out range object, which has a default value of 10
88
89 # Print out variable count, which has the value 10
90
91 # Print out range object, which has a default value of 10
92
93 # Print out variable count, which has the value 10
94
95 # Print out range object, which has a default value of 10
96
97 # Print out variable count, which has the value 10
98
99 # Print out range object, which has a default value of 10
100

```

Run or submit code when you're ready.