

AlgoExpert

Quad Layout

Java

12px

Sublime

Monok

PromptScratchpadOur Solution(s)Video ExplanationRun Code

Solution 1Solution 2Solution 3

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

2

3import java.util.*;

4

5class Program {

6// Average case: when the created BST is balanced

7// O(nlog(n)) time | O(n) space - where n is the length of the array

8// ---

9// Worst case: when the the created BST is like a linked list

10// O(n^2) time | O(n) space

11public static List<Integer> rightSmallerThan(List<Integer> array) {

12if (array.size() == 0) return new ArrayList<Integer>();

13

14List<Integer> rightSmallerCounts = new ArrayList<Integer>(array);

15int lastIdx = array.size() - 1;

16SpecialBST bst = new SpecialBST(array.get(lastIdx));

17rightSmallerCounts.set(lastIdx, 0);

18for (int i = array.size() - 2; i >= 0; i--) {

19bst.insert(array.get(i), i, rightSmallerCounts);

20}

21return rightSmallerCounts;

22}

23

24static class SpecialBST {

25public int value;

26public int leftSubtreeSize;

27public SpecialBST left;

28public SpecialBST right;

29

30public SpecialBST(int value) {

31this.value = value;

32leftSubtreeSize = 0;

33left = null;

34right = null;

35}

36

37public void insert(int value, int idx, List<Integer> rightSmallerCounts) {

38insertHelper(value, idx, rightSmallerCounts, 0);

39}

40

41public void insertHelper(int value, int idx, List<Integer> rightSmallerCounts, int numSmallerAtInsertTime) {

42if (value < this.value) {

43leftSubtreeSize++;

44if (left == null) {

45left = new SpecialBST(value);

46rightSmallerCounts.set(idx, numSmallerAtInsertTime);

47} else {

48left.insertHelper(value, idx, rightSmallerCounts, numSmallerAtInsertTime);

49}

50} else {

51numSmallerAtInsertTime += leftSubtreeSize;

52if (value > this.value)

53numSmallerAtInsertTime++;

54if (right == null) {

55right = new SpecialBST(value);

56rightSmallerCounts.set(idx, numSmallerAtInsertTime);

57} else {

58right.insertHelper(value, idx, rightSmallerCounts, numSmallerAtInsertTime);

59}

60}

61}

62}

63}

64

