

Solution 1

Solution 2

Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 ▼ import java.util.*;
4
5 ▼ class 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
11    ▼ public static List<Integer> rightSmallerThan(List<Integer> array) {
12        if (array.size() == 0) return new ArrayList<Integer>();
13
14        int lastIdx = array.size() - 1;
15        SpecialBST bst = new SpecialBST(array.get(lastIdx), lastIdx, 0);
16    ▼ for (int i = array.size() - 2; i >= 0; i--) {
17        bst.insert(array.get(i), i);
18    }
19
20    List<Integer> rightSmallerCounts = new ArrayList<Integer>(array);
21    getRightSmallerCounts(bst, rightSmallerCounts);
22    return rightSmallerCounts;
23    }
24
25    ▼ public static void getRightSmallerCounts(SpecialBST bst, List<Integer> rightSmallerCounts) {
26        if (bst == null) return;
27        rightSmallerCounts.set(bst.idx, bst.numSmallerAtInsertTime);
28        getRightSmallerCounts(bst.left, rightSmallerCounts);
29        getRightSmallerCounts(bst.right, rightSmallerCounts);
30    }
31
32    ▼ static class SpecialBST {
33        public int value;
34        public int idx;
35        public int numSmallerAtInsertTime;
36        public int leftSubtreeSize;
37        public SpecialBST left;
38        public SpecialBST right;
39
40    ▼ public SpecialBST(int value, int idx, int numSmallerAtInsertTime) {
41        this.value = value;
42        this.idx = idx;
43        this.numSmallerAtInsertTime = numSmallerAtInsertTime;
44        leftSubtreeSize = 0;
45        left = null;
46        right = null;
47    }
48
49    ▼ public void insert(int value, int idx) {
50        insertHelper(value, idx, 0);
51    }
52
53    ▼ public void insertHelper(int value, int idx, int numSmallerAtInsertTime) {
54    ▼ if (value < this.value) {
55        leftSubtreeSize++;
56    ▼ if (left == null) {
57        left = new SpecialBST(value, idx, numSmallerAtInsertTime);
58    ▼ } else {
59        left.insertHelper(value, idx, numSmallerAtInsertTime);
60    }
61    ▼ } else {
62        numSmallerAtInsertTime += leftSubtreeSize;
63        if (value > this.value)
64            numSmallerAtInsertTime++;
65    ▼ if (right == null) {
66        right = new SpecialBST(value, idx, numSmallerAtInsertTime);
67    ▼ } else {
68        right.insertHelper(value, idx, numSmallerAtInsertTime);
69    }
70    }
71    }
72    }
73    }
74    }
```

