12px Sublime Monokai 00:00:00 AlgoExpert **Quad Layout** Java

Scratchpad Our Solution(s) Video Explanation Run Code

```
Solution 1
```

Prompt

43

```
1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
 3 import java.util.*;
 5 class Program {
     // O(n) time | O(d) space - where n is the number of nodes in
      // the Binary Tree and d is the depth (height) of the Binary Tree
      public static BinaryTree rightSiblingTree(BinaryTree root) {
        mutate(root, null, false);
10
        return root;
11
12
13
      public static void mutate(BinaryTree node, BinaryTree parent, boolean isLeftChild) {
14
        if (node == null) return;
15
        var left = node.left;
var right = node.right;
16
17
        mutate(left, node, true);
18
19
        if (parent == null) {
        node.right = null;
} else if (isLeftChild) {
20
21
22
          node.right = parent.right;
23
        } else {
24
          if (parent.right == null) {
25
           node.right = null;
26
          } else {
27
28
29
            node.right = parent.right.left;
30
31
        mutate(right, node, false);
32
33
      static class BinaryTree {
34
        int value;
35
        BinaryTree left = null;
        BinaryTree right = null;
36
37
38
        public BinaryTree(int value) {
          this.value = value;
39
40
41
42 }
```