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

Run Code Scratchpad Our Solution(s) Video Explanation

```
Solution 1 Solution 2
```

Prompt

```
1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
 3 import java.util.*;
 5 class Program {
     // O(n) time \mid O(n) space - where n is the number of nodes in the Binary Tree
      public static BinaryTree flattenBinaryTree(BinaryTree root) {
        List<BinaryTree> inOrderNodes = getNodesInOrder(root, new ArrayList<BinaryTree>());
        for (int i = 0; i < inOrderNodes.size() - 1; i++) {</pre>
10
          BinaryTree leftNode = inOrderNodes.get(i);
11
          BinaryTree rightNode = inOrderNodes.get(i + 1);
12
          leftNode.right = rightNode;
13
          rightNode.left = leftNode;
14
15
        return inOrderNodes.get(0);
16
17
18
      public static List<BinaryTree> getNodesInOrder(BinaryTree tree, List<BinaryTree> array) {
        if (tree != null) {
19
20
21
          getNodesInOrder(tree.left, array);
          array.add(tree);
22
          getNodesInOrder(tree.right, array);
23
24
        return array;
25
26
27
28
29
      static class BinaryTree {
        int value;
        BinaryTree left = null;
30
        BinaryTree right = null;
31
32
        public BinaryTree(int value) {
33
         this.value = value;
34
35
36 }
```