

AlgoExpert

Quad Layout

Java

12px

Sublime

Monokai

00:00:00

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2

1// Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3import java.util.*;
4
5class Program {
6// O(n) time | O(d) space - where n is the number of nodes in the Binary Tree
7// and d is the depth (height) of the Binary Tree
8public static BinaryTreeNode flattenBinaryTree(BinaryTreeNode root) {
9flattenTree(root);
10return getLeftMost(root);
11}
12
13public static BinaryTreeNode[] flattenTree(BinaryTreeNode node) {
14BinaryTreeNode leftMost;
15BinaryTreeNode rightMost;
16
17if (node.left == null) {
18leftMost = node;
19} else {
20BinaryTreeNode[] leftAndRightMostNodes = flattenTree(node.left);
21connectNodes(leftAndRightMostNodes[1], node);
22leftMost = leftAndRightMostNodes[0];
23}
24
25if (node.right == null) {
26rightMost = node;
27} else {
28BinaryTreeNode[] leftAndRightMostNodes = flattenTree(node.right);
29connectNodes(node, leftAndRightMostNodes[0]);
30rightMost = leftAndRightMostNodes[1];
31}
32
33return new BinaryTreeNode[] {leftMost, rightMost};
34}
35
36public static void connectNodes(BinaryTreeNode left, BinaryTreeNode right) {
37left.right = right;
38right.left = left;
39}
40
41public static BinaryTreeNode getLeftMost(BinaryTreeNode node) {
42while (node.left != null) {
43node = node.left;
44}
45return node;
46}
47
48static class BinaryTreeNode {
49int value;
50BinaryTreeNode left = null;
51BinaryTreeNode right = null;
52
53public BinaryTreeNode(int value) {
54this.value = value;
55}
56}
57}
58

