

AlgoExpert

Quad Layout

C#

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

