AlgoExpert Quad Layout Python 12px Sublime Monokai 00:00:00

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

```
Solution 1 Solution 2
```

38

```
_{\rm 1} \, # Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
    class BinaryTree:
        def __init__(self, value, left=None, right=None):
            self.value = value
            self.left = left
            self.right = right
10 \, # O(n) time | O(d) space - where n is the number of nodes in the Binary Tree
11\, \, # and d is the depth (height) of the Binary Tree \,
12 def flattenBinaryTree(root):
        leftMost, _ = flattenTree(root)
return leftMost
13
14
15
16
    def flattenTree(node):
17
18
        if node.left is None:
19
            leftMost = node
20
21
        else:
            leftSubtreeLeftMost, leftSubtreeRightMost = flattenTree(node.left)
22
             {\tt connectNodes(leftSubtreeRightMost, node)}
23
            leftMost = leftSubtreeLeftMost
24
25
26
        if node.right is None:
            rightMost = node
27
28
29
        else:
            rightSubtreeLeftMost, rightSubtreeRightMost = flattenTree(node.right)
             {\tt connectNodes(node, rightSubtreeLeftMost)}
30
            rightMost = rightSubtreeRightMost
31
32
        return [leftMost, rightMost]
33
34
35 def connectNodes(left, right):
36
        left.right = right
         right.left = left
37
```