AlgoExpert

Solution 1

Quad Layout

Python

12px

Sublime

Monokai

00:00:

Run Code

Our Solution(s)

```
Run Code
```

Your Solutions

Solution 1 Solution 2 Solution 3

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
    class BinaryTree:
        def __init__(self, value):
            self.value = value
            self.left = None
            self.right = None
    \# O(n) time | O(n) space - where n is the number of nodes in the Binary Tree
    def branchSums(root):
12
        sums = []
13
        calculateBranchSums(root, 0, sums)
14
        return sums
16
    def calculateBranchSums(node, runningSum, sums):
17
18
        if node is None:
19
           return
20
        newRunningSum = runningSum + node.value
22
        if node.left is None and node.right is None:
           sums.append(newRunningSum)
24
           return
25
26
        \verb|calculateBranchSums(node.left, newRunningSum, sums)|\\
        calculateBranchSums(node.right, newRunningSum, sums)
27
```

```
\ensuremath{\mathtt{1}} \ensuremath{\mathtt{\#}} This is the class of the input root. Do not edit it.
   class BinaryTree:
        def __init__(self, value):
             self.value = value
             self.left = None
             self.right = None
   def branchSums(root):
        # Write your code here.
        pass
```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.

Our Tests