

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class BinaryTree:
4     def __init__(self, value):
5         self.value = value
6         self.left = None
7         self.right = None
8
9
10 # O(n) time | O(n) space - where n is the number of nodes in the Binary Tree
11 def branchSums(root):
12     sums = []
13     calculateBranchSums(root, 0, sums)
14     return sums
15
16
17 def calculateBranchSums(node, runningSum, sums):
18     if node is None:
19         return
20
21     newRunningSum = runningSum + node.value
22     if node.left is None and node.right is None:
23         sums.append(newRunningSum)
24         return
25
26     calculateBranchSums(node.left, newRunningSum, sums)
27     calculateBranchSums(node.right, newRunningSum, sums)
28
```

Solution 1Solution 2Solution 3

```
1 # This is the class of the input root. Do not edit it.
2 class BinaryTree:
3     def __init__(self, value):
4         self.value = value
5         self.left = None
6         self.right = None
7
8
9 def branchSums(root):
10     # Write your code here.
11     pass
12
```

**Run or submit code when you're ready.**

Our Tests