

Our Solution(s)

Run Code

Solution 1

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class BinaryTreeNode:
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
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
```

Our Tests

```
1 # Test 1
2 # Input: root = BinaryTreeNode(1)
3 # Output: [1]
4
5 # Test 2
6 # Input: root = BinaryTreeNode(1, BinaryTreeNode(2), BinaryTreeNode(3))
7 # Output: [1, 3, 4]
8
9 # Test 3
10 # Input: root = BinaryTreeNode(1, BinaryTreeNode(2, BinaryTreeNode(4)), BinaryTreeNode(3, BinaryTreeNode(5)))
11 # Output: [1, 6, 7, 9]
```

Your Solutions

Run Code

Solution 1 Solution 2 Solution 3

```
1 # This is the class of the input root. Do not edit it.
2 class BinaryTreeNode:
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
```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.