

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 type BinaryTreeNode struct {
6     Value int
7     Left *BinaryTreeNode
8     Right *BinaryTreeNode
9 }
10
11 // O(n) time | O(n) space - where n is the number of nodes in the Binary Tree
12 func BranchSums(root *BinaryTreeNode) []int {
13     sums := []int{}
14     calculateBranchSums(root, 0, &sums)
15     return sums
16 }
17
18 func calculateBranchSums(node *BinaryTreeNode,
19     runningSum int, sums *[]int) {
20     if node == nil {
21         return
22     }
23
24     runningSum += node.Value
25     if node.Left == nil && node.Right == nil {
26         *sums = append(*sums, runningSum)
27         return
28     }
29
30     calculateBranchSums(node.Left, runningSum, sums)
31     calculateBranchSums(node.Right, runningSum, sums)
32 }
33
```

Solution 1

Solution 2

Solution 3

```
1 package main
2
3 // This is the struct of the input root. Do not edit it.
4 type BinaryTreeNode struct {
5     Value int
6     Left *BinaryTreeNode
7     Right *BinaryTreeNode
8 }
9
10 func BranchSums(root *BinaryTreeNode) []int {
11     // Write your code here.
12     return nil
13 }
14
```

Our Tests

Custom Output

Submit Code

```
18 Run in Jupyter Notebook Notebook 2
19 tree = RandomForestClassifier
20 classifier = RandomForest
21 model = GradientDescent
22 model.fit(train, test)
23
24
25 Run in Jupyter Notebook Notebook 2
26 tree = RandomForestClassifier
27 classifier = RandomForest
28 model = GradientDescent
29 model.fit(train, test)
30
31
```

Run or submit code when you're ready.