Our Solution(s) Run Code

Solution 1

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
    class BST {
        var value: Int
        var left: BST?
        var right: BST?

        init(value: Int) {
            self.value = value
        }
    }

    // O(n) time | O(n) space - where n is the number of nodes in the Binary Tree
    func branchSums(root: BST) -> [Int] {
        var sums = [Int]()
        calculateBranchSums(node: root, runningSum: 0, sums: &sums)
        return sums
    }

    func calculateBranchSums(node: BST?, runningSum: Int, sums: inout [Int]) {
        if let n = node {
            let newRunningSum = runningSum + n.value
            if n.left == nil, n.right == nil {
                sums.append(newRunningSum)
                return
            }
            calculateBranchSums(node: n.left, runningSum: newRunningSum, sums: &su
            calculateBranchSums(node: n.right, runningSum: newRunningSum, sums: &s
        }
    }
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```swift
class Program {
    class BST {
        var value: Int
        var left: BST?
        var right: BST?

        init(value: Int) {
            self.value = value
        }
    }

    func branchSums(root: BST) -> [Int] {
        // Write your code here.
        return []
    }
}

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.