Our Solution(s) Run Code

Solution 1

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

class BinaryTree {
  constructor(value) {
    this.value = value;
    this.left = null;
    this.right = null;
  }
}

// O(n) time | O(n) space - where n is the number of nodes in the Bina
function branchSums(root) {
  const sums = [];
  calculateBranchSums(root, 0, sums);
  return sums;
}

function calculateBranchSums(node, runningSum, sums) {
  if (!node) return;

  const newRunningSum = runningSum + node.value;
  if (!node.left && !node.right) {
    sums.push(newRunningSum);
    return;
  }

  calculateBranchSums(node.left, newRunningSum, sums);
  calculateBranchSums(node.right, newRunningSum, sums);
}

exports.BinaryTree = BinaryTree;
exports.branchSums = branchSums;

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
// This is the class of the input root.
// Do not edit it.
class BinaryTree {
  constructor(value) {
    this.value = value;
    this.left = null;
    this.right = null;
  }
}

function branchSums(root) {
  // Write your code here.
}

// Do not edit the lines below.
exports.BinaryTree = BinaryTree;
exports.branchSums = branchSums;

```

Custom Output Submit Code

Run or submit code when you're ready.