Our Solution(s) Run Code

Solution 1

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

using System;
using System.Collections.Generic;

public class Program {
  // O(n) time | O(log(n)) space
  public static int MaxPathSum(BinaryTree tree) {
    List<int> maxSumArray = findMaxSum(tree);
    return maxSumArray[1];
  }

  public static List<int> findMaxSum(BinaryTree tree) {
    if (tree == null) {
      return new List<int>(){
              0, 0
      };
    }
    List<int> leftMaxSumArray = findMaxSum(tree.left);
    int leftMaxSumAsBranch = leftMaxSumArray[0];
    int leftMaxPathSum = leftMaxSumArray[1];

    List<int> rightMaxSumArray = findMaxSum(tree.right);
    int rightMaxSumAsBranch = rightMaxSumArray[0];
    int rightMaxPathSum = rightMaxSumArray[1];

    int maxChildSumAsBranch = Math.Max(leftMaxSumAsBranch, rightMaxSum
    int maxSumAsBranch = Math.Max(maxChildSumAsBranch + tree.value, tr
    int maxSumAsRootNode = Math.Max(
      leftMaxSumAsBranch + tree.value + rightMaxSumAsBranch,
      maxSumAsBranch
      );
    int maxPathSum = Math.Max(leftMaxPathSum, Math.Max(rightMaxPathSum
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
public class Program {
  public static int MaxPathSum(BinaryTree tree) {
    // Write your code here.
    return -1;
  }

  public class BinaryTree {
    public int value;
    public BinaryTree left;
    public BinaryTree right;

    public BinaryTree(int value) {
      this.value = value;
    }
  }
}

```

Our Tests

Custom Output Submit Code

```
    Utils.AssertTrue(Program.MaxPathSum(test) == 9);
  }

  [Test]
  public void Test3Level() {
    TestBinaryTree test = new TestBinaryTree();
    test.InsertTriple(int[] {1,  2,  3});
    Utils.AssertTrue(Program.MaxPathSum(test) == 5);
  }

  [Test]
  public void Test4Level() {
    TestBinaryTree test = new TestBinaryTree();
    test.InsertTriple(int[] {-5,  3,  4});
    Utils.AssertTrue(Program.MaxPathSum(test) == 9);
```

Run or submit code when you're ready.