AlgoExpert Quad Layout Java 14px Sublime Monokai 00:00:

Our Solution(s) Run Code

Solution 1

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

import java.util.*;

class Program {
  // O(n) time | O(log(n)) space
  public static int maxPathSum(BinaryTree tree) {
    List<Integer> maxSumArray = findMaxSum(tree);
    return maxSumArray.get(1);
  }

  public static List<Integer> findMaxSum(BinaryTree tree) {
    if (tree == null) {
      return new ArrayList<Integer>(Arrays.asList(0, 0));
    }
    List<Integer> leftMaxSumArray = findMaxSum(tree.left);
    Integer leftMaxSumAsBranch = leftMaxSumArray.get(0);
    Integer leftMaxPathSum = leftMaxSumArray.get(1);

    List<Integer> rightMaxSumArray = findMaxSum(tree.right);
    Integer rightMaxSumAsBranch = rightMaxSumArray.get(0);
    Integer rightMaxPathSum = rightMaxSumArray.get(1);

    Integer maxChildSumAsBranch = Math.max(leftMaxSumAsBranch, rightMa
    Integer maxSumAsBranch = Math.max(maxChildSumAsBranch + tree.value
    Integer maxSumAsRootNode =
        Math.max(leftMaxSumAsBranch + tree.value + rightMaxSumAsBranch
    int maxPathSum = Math.max(leftMaxPathSum, Math.max(rightMaxPathSum

    return new ArrayList<Integer>(Arrays.asList(maxSumAsBranch, maxPat
  }

  static class BinaryTree {
```

Your Solutions Run Code

Solution 1 **Solution 2** **Solution 3**

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

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

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

```

Our Tests

[illegible]

Custom Output Submit Code

[illegible]

Run or submit code when you're ready.