Our Solution(s)

Run Code

Solution 1

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(n) time | O(log(n)) space
def maxPathSum(tree):
    _, maxSum = findMaxSum(tree)
    return maxSum


def findMaxSum(tree):
    if tree is None:
        return (0, 0)

    leftMaxSumAsBranch, leftMaxPathSum = findMaxSum(tree.left)
    rightMaxSumAsBranch, rightMaxPathSum = findMaxSum(tree.right)
    maxChildSumAsBranch = max(leftMaxSumAsBranch, rightMaxSumAsBranch

    value = tree.value
    maxSumAsBranch = max(maxChildSumAsBranch + value, value)
    maxSumAsRootNode = max(leftMaxSumAsBranch + value + rightMaxSumAs
    maxPathSum = max(leftMaxPathSum, rightMaxPathSum, maxSumAsRootNod

    return (maxSumAsBranch, maxPathSum)
```

Your Solutions

Run Code

Solution 1   Solution 2   Solution 3

```python
def maxPathSum(tree):
    # Write your code here.
    pass
```

Our Tests

Custom Output

Submit Code

```python
def maxPathSum(tree):
```

```
              self.assertEqual(program.maxPathSum(root), 6)

    def test_case_2(self):
        root = BinaryTree().insert([1, -1])
        self.assertEqual(program.maxPathSum(root), 1)

    def test_case_3(self):
        root = BinaryTree().insert([1, 2, 3])
        self.assertEqual(program.maxPathSum(root), 6)

    def test_case_4(self):
        root = BinaryTree().insert([1, 2, 3, 4, 5])
        self.assertEqual(program.maxPathSum(root), 18)

    def test_case_5(self):
```