

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // O(n) time | O(log(n)) space
4 function maxPathSum(tree) {
5   const [_, maxSum] = [...findMaxSum(tree)];
6   return maxSum;
7 }
8
9 function findMaxSum(tree) {
10   if (tree === null) return [0, 0];
11
12   const [leftMaxSumAsBranch, leftMaxPathSum] = findMaxSum(tree.left);
13   const [rightMaxSumAsBranch, rightMaxPathSum] = findMaxSum(tree.right);
14   const maxChildSumAsBranch = Math.max(leftMaxSumAsBranch, rightMaxSumAsBranch);
15
16   const {value} = tree;
17   const maxSumAsBranch = Math.max(maxChildSumAsBranch + value, value);
18   const maxSumAsRootNode = Math.max(leftMaxSumAsBranch + value + rightMaxSumAsBranch, maxSumAsBranch + value);
19   const maxPathSum = Math.max(leftMaxPathSum, rightMaxPathSum, maxSumAsRootNode);
20
21   return [maxSumAsBranch, maxPathSum];
22 }
23
24 exports.maxPathSum = maxPathSum;
25
```

Solution 1

Solution 2

Solution 3

```
1 function maxPathSum(tree) {
2   // Write your code here.
3 }
4
5 // Do not edit the line below.
6 exports.maxPathSum = maxPathSum;
7
```

Our Tests

Custom Output

Submit Code

```
1 // Test Case 1: Empty tree
2 const tree = null;
3
4 const result = maxPathSum(tree);
5
6 console.log(result); // Expected: 0
7
8 // Test Case 2: Single node
9 const tree = {
10   value: 1,
11   left: null,
12   right: null
13 };
14
15 const result = maxPathSum(tree);
16
17 console.log(result); // Expected: 1
18
19 // Test Case 3: Two nodes
20 const tree = {
21   value: 1,
22   left: {
23     value: 2,
24     left: null,
25     right: null
26   },
27   right: null
28 };
29
30 const result = maxPathSum(tree);
31
32 console.log(result); // Expected: 3
33
```

```
1 // Test Case 4: Three nodes
2 const tree = {
3   value: 1,
4   left: {
5     value: 2,
6     left: {
7       value: 3,
8       left: null,
9       right: null
10    },
11    right: null
12  },
13  right: null
14 };
15
16 const result = maxPathSum(tree);
17
18 console.log(result); // Expected: 6
19
```

Custom Output

Submit Code

```
10 int main() {
11     // Create a vector of integers
12     vector<int> v;
13     // Add elements to the vector
14     v.push_back(10);
15     v.push_back(20);
16     v.push_back(30);
17     // Print the size of the vector
18     cout << v.size() << endl;
19     // Print the elements of the vector
20     for (int i = 0; i < v.size(); i++) {
21         cout << v[i] << " ";
22     }
23     return 0;
24 }
```

Run or submit code when you're ready.