

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);
19  const maxPathSum = Math.max(leftMaxPathSum, rightMaxPathSum, maxSumAsRootNode);
20
21  return [maxSumAsBranch, maxPathSum];
22 }
23
24 exports.maxPathSum = maxPathSum;
25
```