

Student: Ovidiu Mura

## Problem 124: Binary Tree Maximum Path Sum

**Success** Details >

Runtime: **0 ms**, faster than **100.00%** of Java  
Tree Maximum Path Sum.

Memory Usage: **41.3 MB**, less than **9.52%** of  
Binary Tree Maximum Path Sum.

Next challenges:

[Path Sum](#)

[Sum Root to Leaf Numbers](#)

[Longest Univalue Path](#)

[Time Needed to In](#)

Show off your acceptance:



Time Submitted	Status	Runtime
a few seconds ago	Accepted	0 ms
3 minutes ago	Accepted	0 ms

```
1  /**
2   * Definition for a binary tree node.
3   * public class TreeNode {
4   *     int val;
5   *     TreeNode left;
6   *     TreeNode right;
7   *     TreeNode(int x) { val = x; }
8   * }
9   */
10 class Solution {
11     class Result {
12         public int val = Integer.MIN_VALUE;
13     }
14     private int maxPathSum(TreeNode root, Result max) {
15         if (root == null)
16             return 0;
17
18         int left = maxPathSum(root.left, max);
19         int right = maxPathSum(root.right, max);
20
21         int current = Math.max(root.val, Math.max(root.val + left, root.val + right));
22
23         max.val = Math.max(max.val, Math.max(current, left + root.val + right));
24
25         return current;
26     }
27
28     public int maxPathSum(TreeNode root) {
29         Result max = new Result();
30         maxPathSum(root, max);
31         return max.val;
32     }
33 }
```