Subtree with Maximum Value

Given a binary tree `root`, return the maximum sum of a subtree. A subtree is defined to be some node in `root` including all of its descendants. A subtree sum is the sum of all the node values in the subtree. A subtree can be null in which case its sum is `0`.

Constraints

- `1 ≤ n ≤ 100,000` where `n` is the number of nodes in `root`
- **Input**

  Visualize

  root = [3, [0, null, null], [2, [0, null, null],null]]

  link:https://binarysearch.com/problems/Subtree-with-Maximum-Value

```
def solve(self, root):
        sm = [0]
        self.helper(root,sm)
        return sm[0]



    def helper(self,root,sm):
        if root is None:
            return 0
        lt = self.helper(root.left,sm)
        rt = self.helper(root.right,sm)
        sm[0] = max(sm[0],lt+rt+root.val)
        return lt+rt+root.val
```