class Solution:

def maxPathSum(self, root: Optional[TreeNode]) -> int:

res = root.val

def dfs(node):

nonlocal res

if not node:

return 0

# recursively compute the maximum sum of the left and right subtree paths

left\_sum = max(0, dfs(node.left))

right\_sum = max(0, dfs(node.right))

# update the maximum path sum encountered so far(with split)

res = max(res, left\_sum + right\_sum + node.val)

# return the maximum sum of the path(without split)

return max(left\_sum, right\_sum) + node.val

dfs(root)

return res