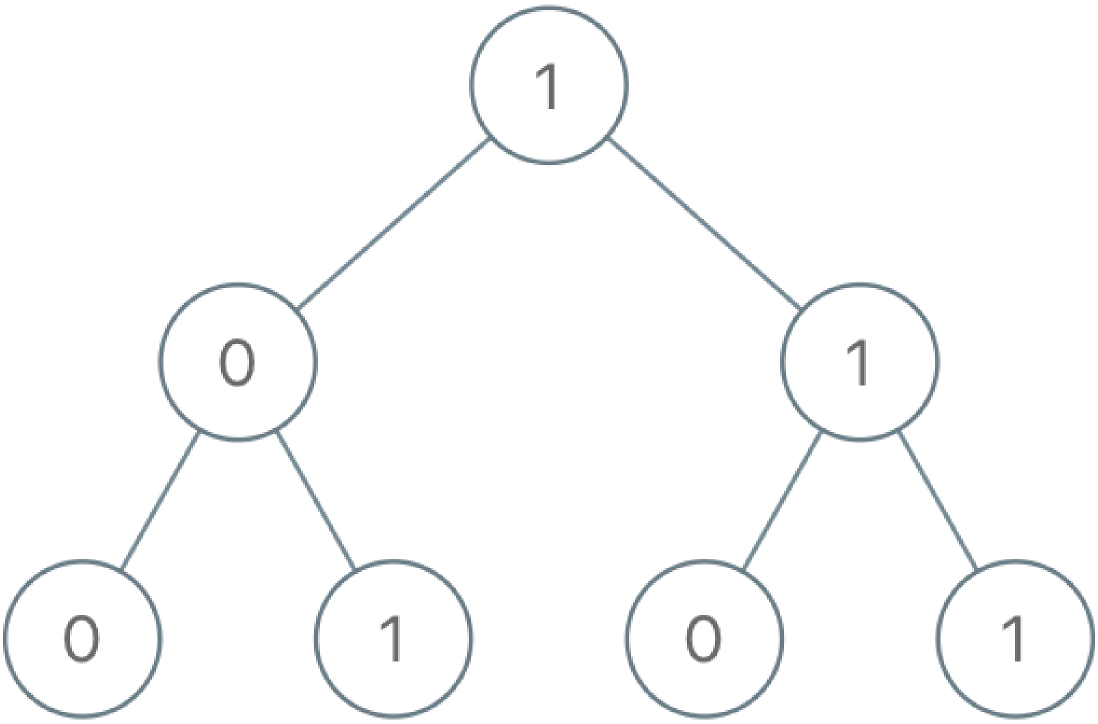
1022. Sum of Root To Leaf Binary Numbers

You are given the `root` of a binary tree where each node has a value `0` or `1`. Each root-to-leaf path represents a binary number starting with the most significant bit. For example, if the path is `0 -> 1 -> 1 -> 0 -> 1`, then this could represent `01101` in binary, which is `13`.

For all leaves in the tree, consider the numbers represented by the path from the root to that leaf.

Return *the sum of these numbers*. The answer is **guaranteed** to fit in a **32-bits** integer.



Input: root = [1,0,1,0,1,0,1]
Output: 22
Explanation: (100) + (101) + (110) + (111) = 4 + 5 + 6 + 7 = 22

```
def sumRootToLeaf(self, root: TreeNode) -> int:
        res = [0]
        ssf = ''
        self.helper(root,res,ssf)
        return res[0]

    def helper(self,root,res,ssf):
```

```python
        if root is None:
            return
        if root.left is None and root.right is None:
            ssf = ssf+str(root.val)
            temp = int(ssf,2)
            res[0] = res[0]+temp
            return
        ssf = ssf+str(root.val)
        self.helper(root.left,res,ssf)
        self.helper(root.right,res,ssf)
```