

404. Sum of Left Leaves

404. Sum of Left Leaves

Easy 3877 262 Add to List Share

Given the `root` of a binary tree, return the sum of all left leaves.

A leaf is a node with no children. A **left leaf** is a leaf that is the left child of another node.

Example 1:

3

9

20

15

7

Input: root = [3,9,20,null,15,7]

Output: 24

Explanation: There are two left leaves in the binary tree, with values 9 and 15 respectively.

Example 2:

Input: root = []

Output: 0

Constraints:

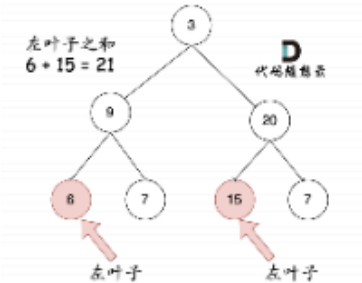
- The number of nodes in the tree is in the range [0, 1000].
- $-1000 \leq \text{Node.val} \leq 1000$

題目

左葉子節點

葉子節點是沒有孩子節點的節點，那左葉子節點就是"是左孩子且該左孩子沒有孩子節點"

節點A的左孩子不為空，且左孩子的左右孩子都為空（說明是葉子節點），那麼A節點的左孩子為左葉子節點



遞迴法

前序遍歷(中, 左, 右)

1. 確定遞迴函數的參數和返回值

判斷一個樹的左葉子節點之和，那麼一定要傳入樹的根節點，遞迴函數的返回值為數值之和，所以為int

使用題目中給出的函數就可以了。

2. 確定終止條件

如果遍歷到空節點，那麼左葉子值一定是0

```
# 遞迴中止條件
if root == None:
    return 0
```

3. 確定單層遞迴的邏輯(找出重複的子問題)

對應到本題是：左葉子節點操作，遞迴左子樹，遞迴右子樹。

所以這個重複的子問題就出來了，先判斷當前節點的左孩子是否為左葉子節點，再遍歷左子樹，最後遍歷右子樹。

```
# 如果遍歷到當前節點的左孩子為左葉節點，則將左孩子的值加入 res
if root.left != None and root.left.left == None and root.left.right == None:
    self.res += root.left.val
# 遍歷左子樹
self.sumOfLeftLeaves(root.left)
# 遍歷右子樹
self.sumOfLeftLeaves(root.right)
```

總結

題目要求左葉子之和，其實是比较繞的，因為不能判斷本節點是不是左葉子節點。

此時就要通過節點的父節點來判斷其左孩子是不是左葉子了。

平時我們解二元樹的題目時，已經習慣了通過節點的左右孩子判斷本節點的屬性，而本題我們要通過節點的父節點判斷本節點的屬性。