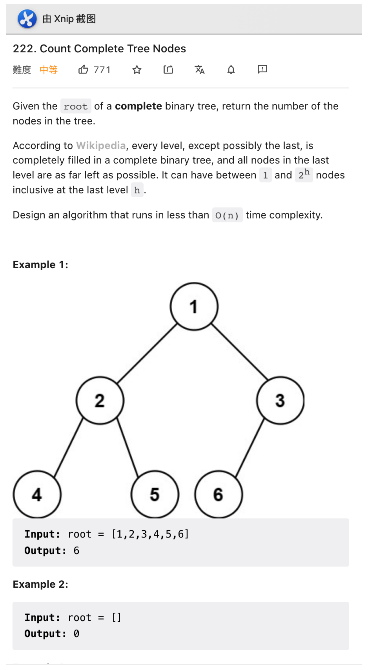
# 222. Count Complete Tree Nodes



由 Xnip 截图

222. Count Complete Tree Nodes

難度 中等 771

Given the `root` of a **complete** binary tree, return the number of the nodes in the tree.

According to Wikipedia, every level, except possibly the last, is completely filled in a complete binary tree, and all nodes in the last level are as far left as possible. It can have between `1` and `2^h` nodes inclusive at the last level `h`.

Design an algorithm that runs in less than `O(n)` time complexity.

**Example 1:**

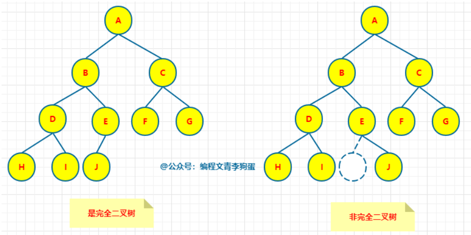
```
Input: root = [1,2,3,4,5,6]
Output: 6
```

**Example 2:**

```
Input: root = []
Output: 0
```

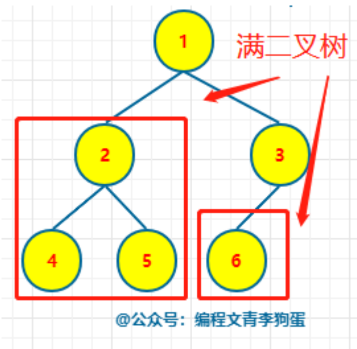
## 完全二元樹的節點個數

- 除了最底層以外，其餘的每一層節點數都是滿的
- 最底層的節點全集中在該層最左邊的位置。
  - 最後一層如果是第 k 層，那麼這一層最少有 1 個葉子節點，最多有 2^(k-1) 個節點，而這些節點都是從左到右依次排列。

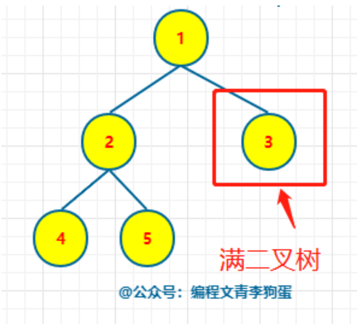


## 對於一棵完全二元樹來說，左子樹的深度必然是 ≥ 右子樹的深度的

- 當左子樹的深度等於右子樹的深度的時候，左子樹必定是滿二元樹。



- 當左子樹的深度大於右子樹的深度的時候，右子樹必定是滿二元樹。



- 當知道子樹是滿二元樹的時候，那這棵子樹的節點數目就是 2^k - 1，其中 k 為子樹的深度

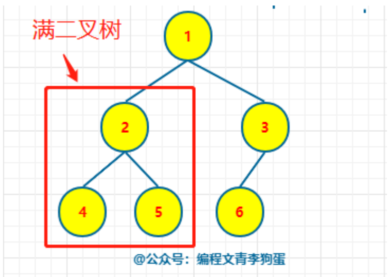
## 題解

- 1.求二元樹的深度

```
def height(self, root:TreeNode):
    height = 0
    while root:
        root = root.left
        height += 1

    return height
```

- 2.左子樹的深度 = 右子樹的深度，左子樹為滿二元樹，直接用公式計算其節點數。
  2 ^ leftHeight - 1


满二叉树
1
2
3
4
5
6
@公众号：编程文青李狗蛋