Problem 222: Count Complete Tree Nodes

Problem Information

Difficulty: Easy
Acceptance Rate: 71.38%
Paid Only: No
Tags: Binary Search, Bit Manipulation, Tree, Binary Tree

Problem Description

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

According to **[Wikipedia](http://en.wikipedia.org/wiki/Binary_tree#Types_of_binary_trees)** , 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 `2h` nodes inclusive at the last level `h`.

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

Example 1:

![](https://assets.leetcode.com/uploads/2021/01/14/complete.jpg)

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

Example 2:

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

Example 3:

Input: root = [1] **Output:** 1

Constraints:

* The number of nodes in the tree is in the range `[0, 5 * 104]`. * `0 <= Node.val <= 5 * 104` * The tree is guaranteed to be **complete**.

Code Snippets

C++:

```
/**
* Definition for a binary tree node.
* struct TreeNode {
* int val;
* TreeNode *left;
* TreeNode *right;
* TreeNode() : val(0), left(nullptr), right(nullptr) {}
* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
* TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
* };
*/
class Solution {
public:
int countNodes(TreeNode* root) {

}
};
```

Java:

```
/**
* Definition for a binary tree node.
* public class TreeNode {
* int val;
* TreeNode left;
* TreeNode right;
* TreeNode() {}
* TreeNode(int val) { this.val = val; }
* TreeNode(int val, TreeNode left, TreeNode right) {
* this.val = val;
* this.left = left;
* this.right = right;
* }
* }
*/
class Solution {
public int countNodes(TreeNode root) {

}
}
```

Python3:

```
# Definition for a binary tree node.
# class TreeNode:
# def __init__(self, val=0, left=None, right=None):
# self.val = val
# self.left = left
# self.right = right
class Solution:
def countNodes(self, root: Optional[TreeNode]) -> int:
```