222. Count Complete Tree Nodes

Description | Hints | Submissions | Discuss | Solution

Pick One

Given a **complete** binary tree, count the number of nodes.

Note:

Definition of a complete binary tree from Wikipedia:
In a complete binary tree every level, except possibly the last, is completely filled, 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.

Example:

```
Input:
    1
   / \
  2   3
 / \  /
4  5 6

Output: 6
```

Seen this question in a real interview before? Yes No

```java
/*
 * 在满二叉树种有一个性质就是节点数是2^h-1,h是高度，所以可以先判断节点的左右高度是不是一致的，如果
 * 是一致的，就说明题目中完全二叉树是满二叉树，就可以用当前的公式，如果左右的高度不相等，然后就用递归进行
 * 计算左右节点。
 */
    public int countNodes(TreeNode root) {
                if (root == null) {
                    return 0;
                }else {
                    int left = getLeftHeight(root);
                    int right = getRightHeight(root);

                    if(left == right)
                        //这是代表1向右移动left，1为2^0,1<<left表示2^left
                        return (1 << left) - 1;
                    else {
                        return countNodes(root.left) + countNodes(root.right) + 1;
                    }
                }
    }
    public static int getRightHeight(TreeNode root) {
        int height = 0;
        while(root != null) {
            height ++;
            root = root.left;
        }
        return height;
    }
    public static int getLeftHeight(TreeNode root) {
        int height = 0;
        while (root != null) {
            height ++;
            root = root.right;
        }
        return height;
    }

}
```