

完全二叉树

2022年9月8日 16:58

完全二叉树的节点个数

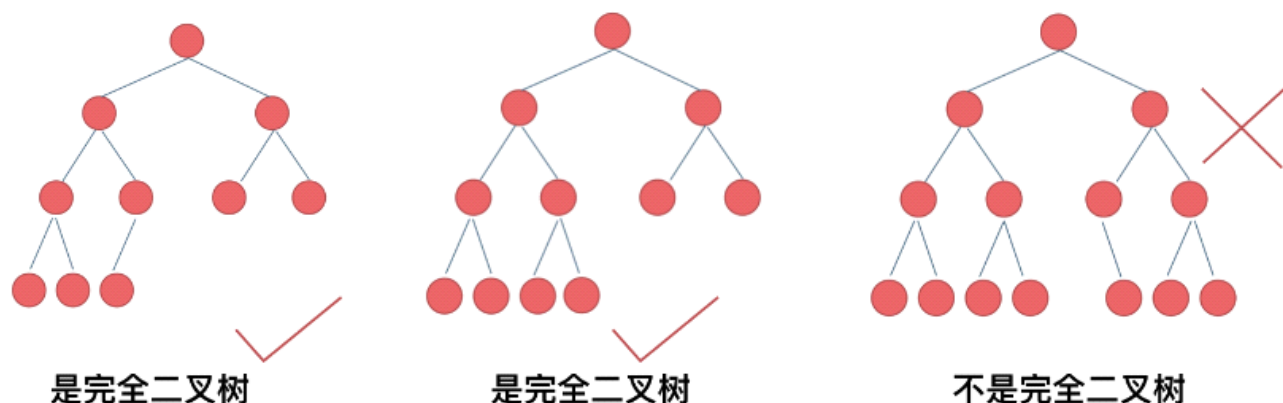
222. 完全二叉树的节点个数

难度 中等 776 ☆ 收藏 通知 评论

给你一棵 **完全二叉树** 的根节点 `root`，求出该树的节点个数。

完全二叉树 的定义如下：在完全二叉树中，除了最底层节点可能没填满外，其余每层节点数都达到最大值，并且最下面一层的节点都集中在该层最左边的若干位置。若最底层为第 h 层，则该层包含 $1 \sim 2^{h-1}$ 个节点。

完全二叉树有一个重要的性质 即：除了最底层节点可能没填满外，其余每层节点数都达到最大值，并且最下面一层都集中在该层最左边的若干位置。若最底层为第 h 层，则该层包含 $1 \sim 2^{h-1}$ 个节点



利用上述性质，我们可以做文章

完全二叉树只有两种情况：

情况一：满二叉树

情况二：最后一层叶子节点没有满

对于情况一 可以直接用 $2^{\text{树深度}-1}$ 来计算。注意这里根节点深度为1

（注意，在c++中计算开方可以使用左移运算符计算 例如 $1 \ll i$ 代表 2^i ）

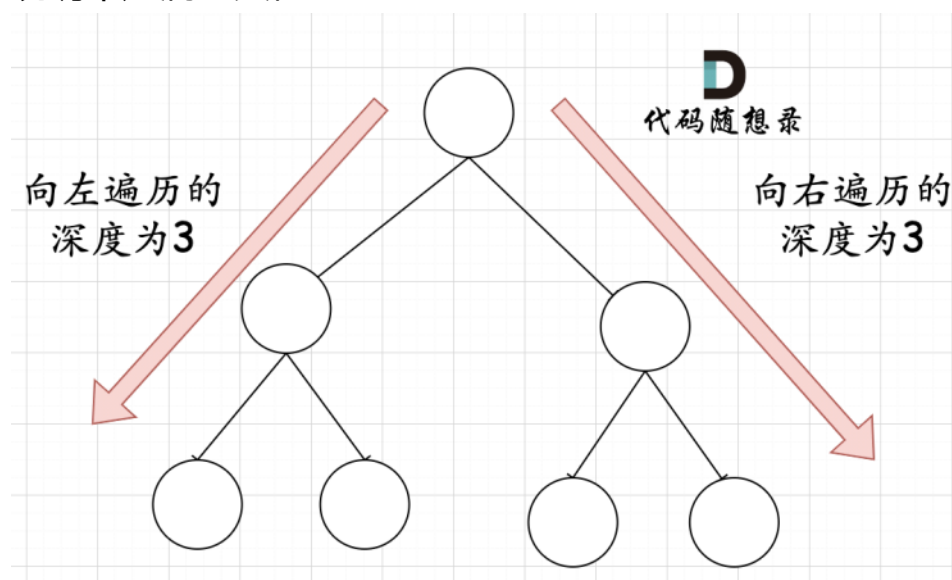
对于情况二，分别递归左孩子和右孩子，递归到某一深度一定会有左孩子

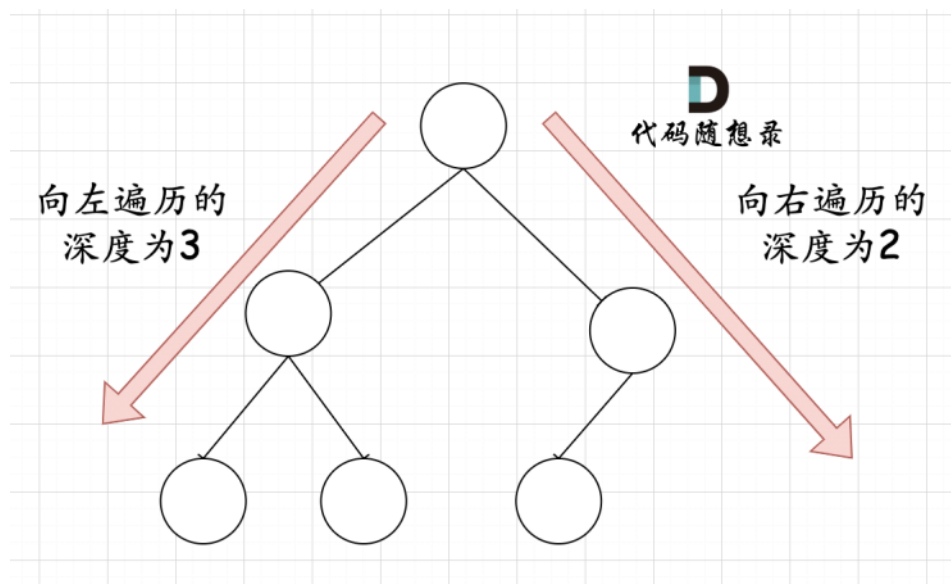
222.完全二叉树的节点个数

代码随想录

满二叉树 满二叉树 满二叉树

在完全二叉树中，若递归向左遍历的深度不等于递归向右遍历的深度，则说明不是满二叉树





上代码，注意通过该方法可以做到 $O(\log n \times \log n)$ 的时间复杂度

```
class Solution {
public:
    int countNodes(TreeNode* root) {
        int leftHigh=countHigh(root,true);
        int rightHigh=countHigh(root,false);
        if(leftHigh==rightHigh)
            return (1 << leftHigh) - 1;
        else
            return 1+(countNodes(root->left)+countNodes(root->right));
    }
    int countHigh(TreeNode* node, bool flag){
        if(!node)
            return 0;
        if(flag){
            return 1+countHigh(node->left,flag);
        }
        else{
            return 1+countHigh(node->right,flag);
        }
    }
};
```