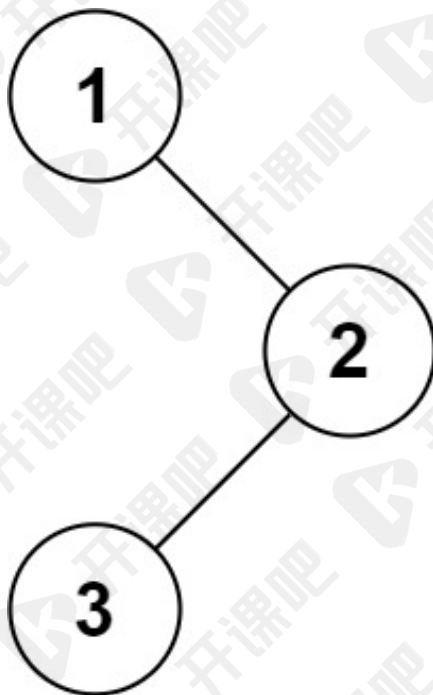


## 【第四周】二叉树（Binary-Tree）与经典问题

### 144. 二叉树的前序遍历

给你二叉树的根节点 `root`，返回它节点值的 前序 遍历。

示例：



```
1 输入: root = [1,null,2,3]
2 输出: [1,2,3]
```

### 589. N 叉树的前序遍历

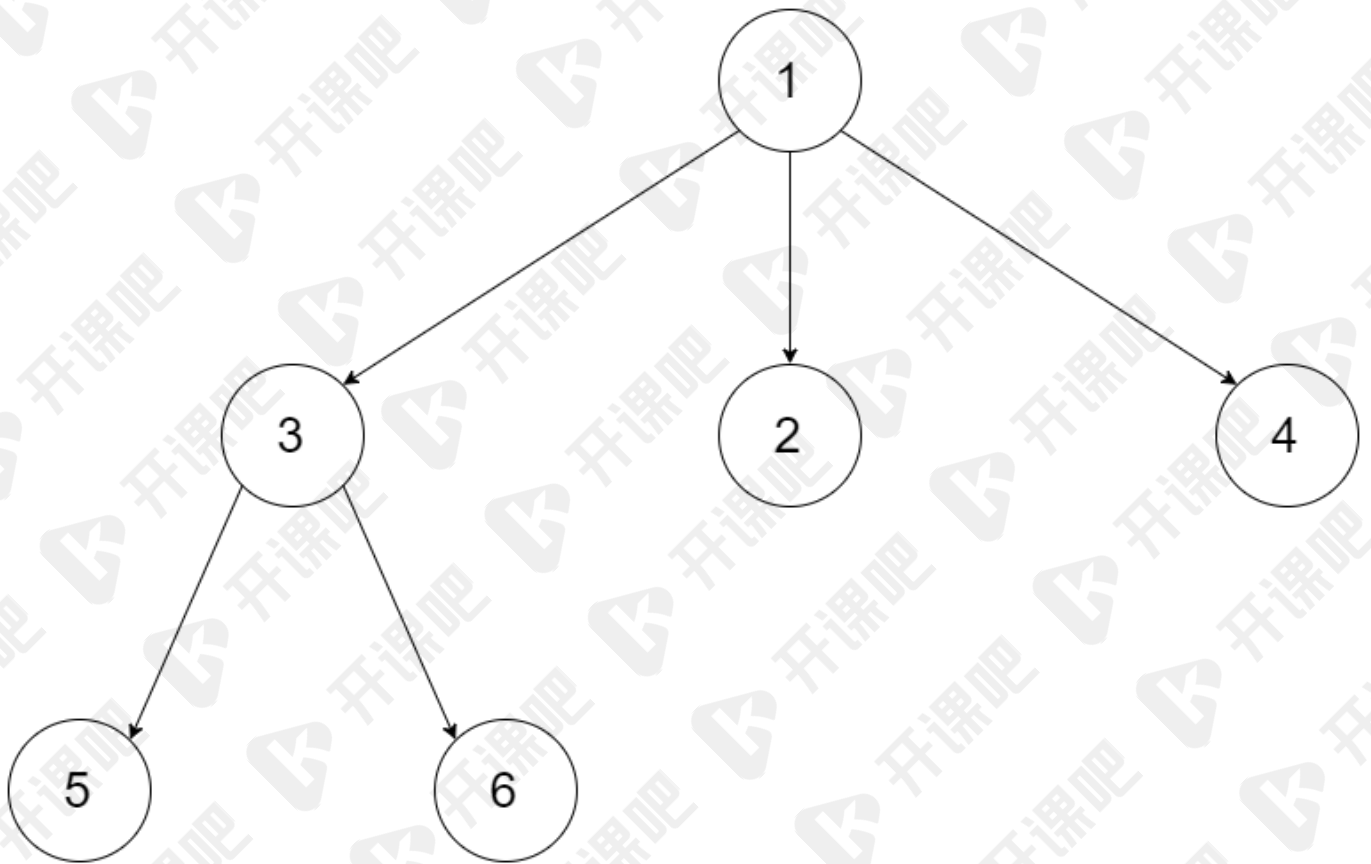
给定一个 N 叉树，返回其节点值的 前序遍历。

N 叉树 在输入中按层序遍历进行序列化表示，每组子节点由空值 `null` 分隔（请参见示例）。

进阶：

递归法很简单，你可以使用迭代法完成此题吗？

示例：



1 输入: root = [1,null,3,2,4,null,5,6]  
2 输出: [1,3,5,6,2,4]

## 226. 翻转二叉树

翻转一棵二叉树。

示例:

输入:

```
1      4
2     / \
3    2   7
4   / \ / \
5  1  3 6  9
```

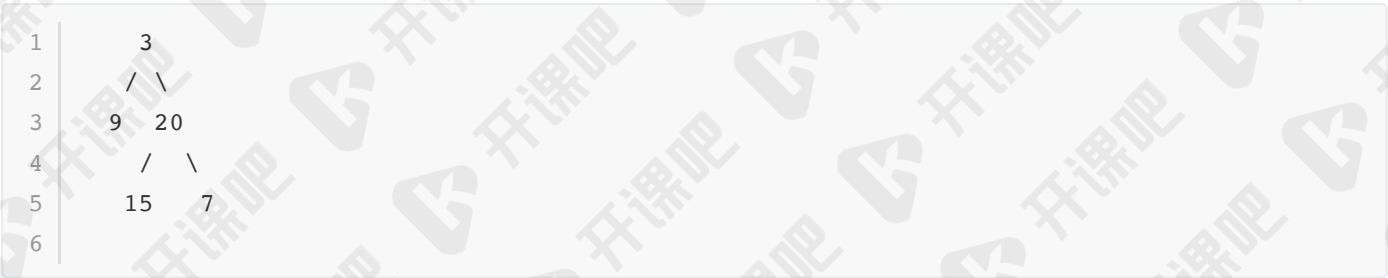
输出:

```
1      4
2     / \
3    7   2
4   / \ / \
5  9  6 3  1
```

## 剑指 Offer 32 - II. 从上到下打印二叉树 II

从上到下按层打印二叉树，同一层的节点按从左到右的顺序打印，每一层打印到一行。

例如:给定二叉树: [3,9,20,null,null,15,7],



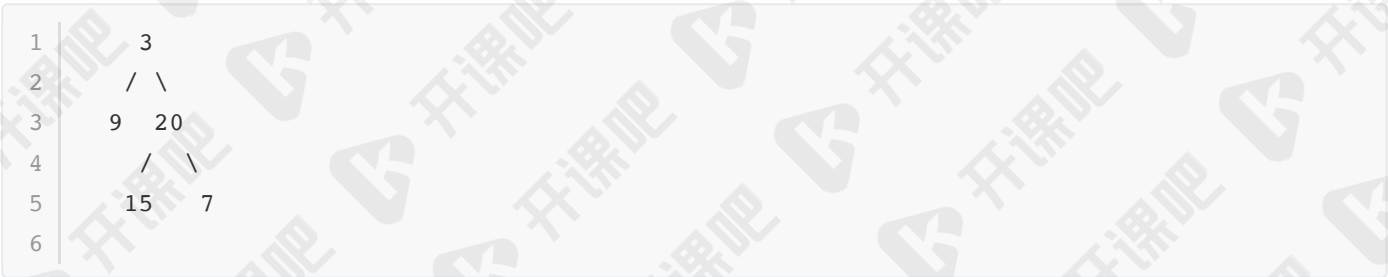
返回其层次遍历结果：



## 107. 二叉树的层序遍历 II

给定一个二叉树，返回其节点值自底向上的层序遍历。（即按从叶子节点所在层到根节点所在的层，逐层从左向右遍历）

例如：给定二叉树 [3,9,20,null,null,15,7],



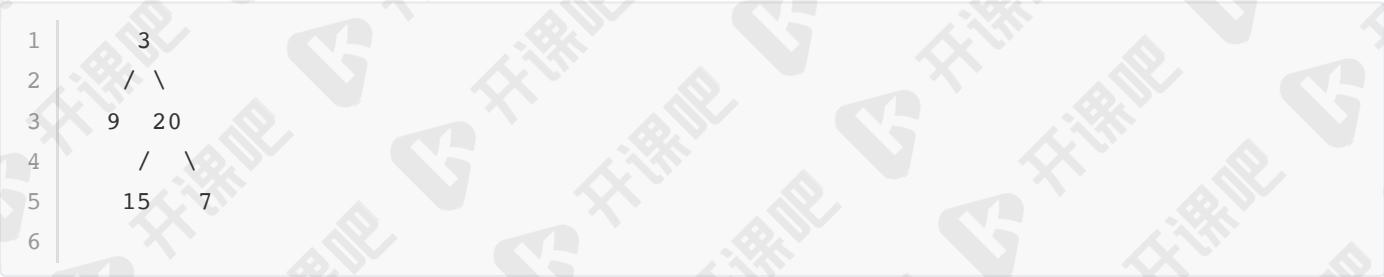
返回其自底向上的层序遍历为：



### 103. 二叉树的锯齿形层序遍历

给定一个二叉树，返回其节点值的锯齿形层序遍历。（即先从左往右，再从右往左进行下一层遍历，以此类推，层与层之间交替进行）。

例如：给定二叉树 [3,9,20,null,null,15,7]，



返回锯齿形层序遍历如下：



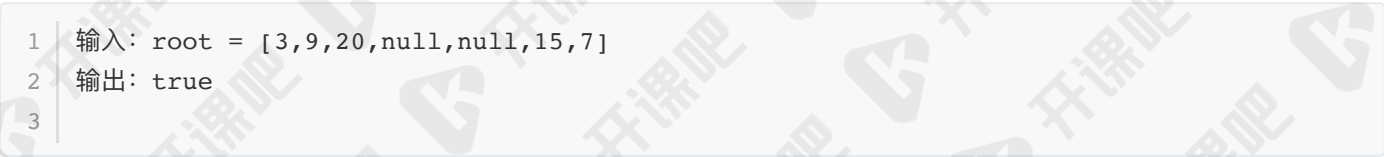
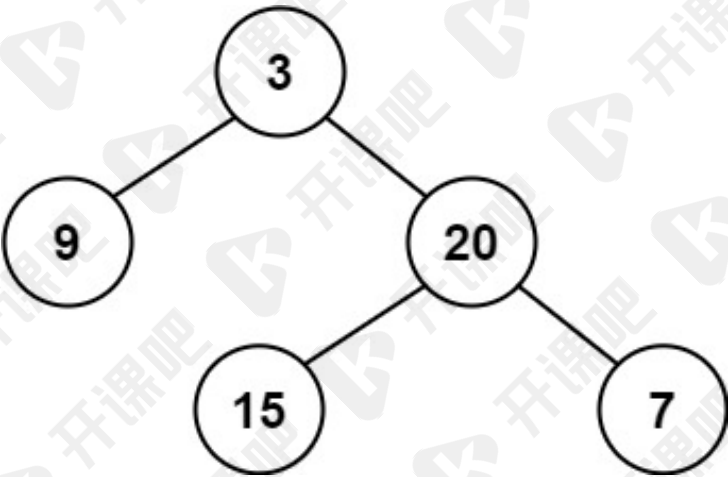
### 110. 平衡二叉树

给定一个二叉树，判断它是否是高度平衡的二叉树。

本题中，一棵高度平衡二叉树定义为：

一个二叉树每个节点 的左右两个子树的高度差的绝对值不超过 1。

示例：

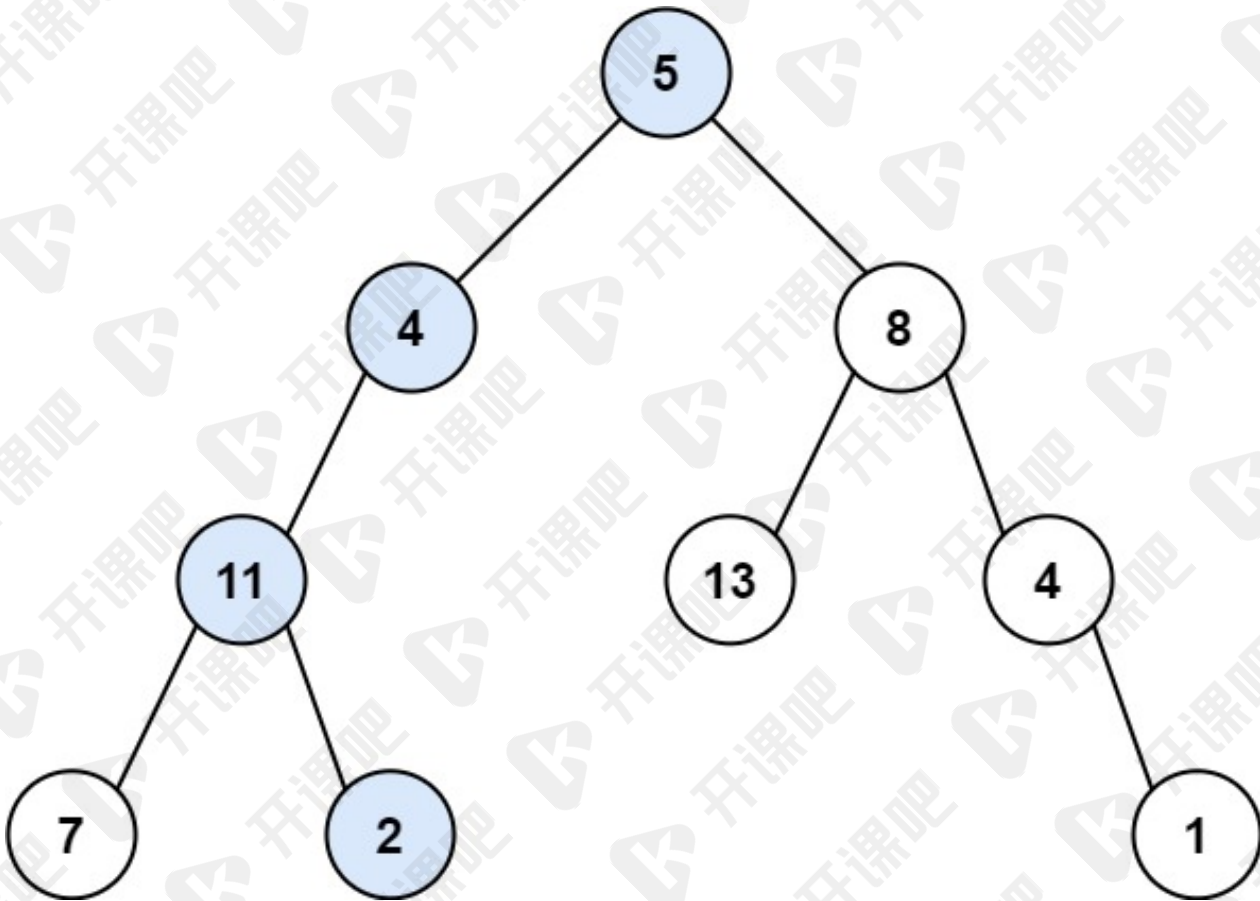


## 112. 路径总和

给你二叉树的根节点 `root` 和一个表示目标和的整数 `targetSum`，判断该树中是否存在 根节点到叶子节点 的路径，这条路径上所有节点值相加等于目标和 `targetSum`。

叶子节点 是指没有子节点的节点。

示例：

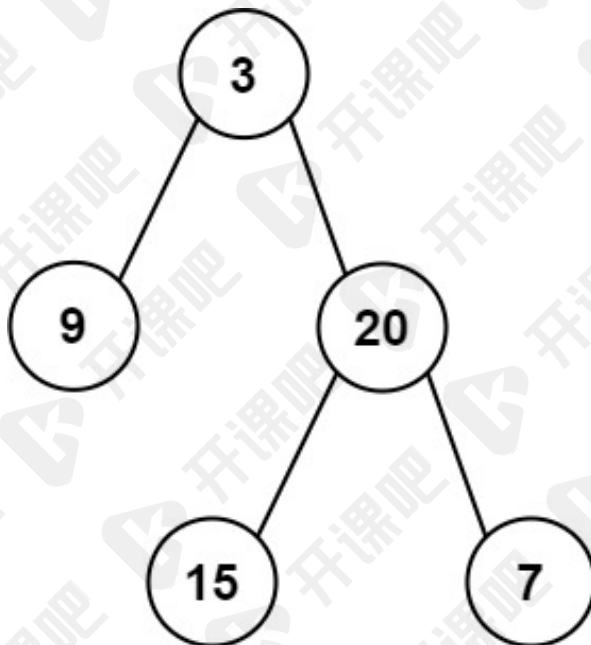


```
1 输入: root = [5,4,8,11,null,13,4,7,2,null,null,null,1], targetSum = 22
2 输出: true
```

## 105. 从前序与中序遍历序列构造二叉树

给定一棵树的前序遍历 `preorder` 与中序遍历 `inorder`。请构造二叉树并返回其根节点。

示例：



```
1 Input: preorder = [3,9,20,15,7], inorder = [9,3,15,20,7]
2 Output: [3,9,20,null,null,15,7]
```

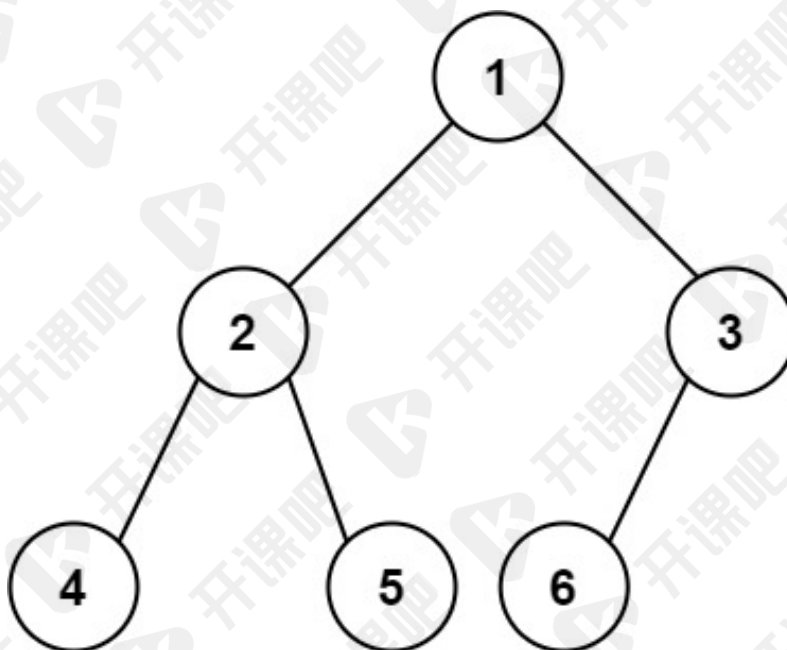
## 222. 完全二叉树的节点个数

难度中等511

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。

示例：



```
1 输入: root = [1,2,3,4,5,6]
2 输出: 6
```



## 剑指 Offer 54. 二叉搜索树的第k大节点

给定一棵二叉搜索树，请找出其中第k大的节点。

示例：

```
1  输入：root = [3,1,4,null,2], k = 1
2      3
3      / \
4     1   4
5      \
6       2
7  输出：4
```

## 剑指 Offer 26. 树的子结构

输入两棵二叉树A和B，判断B是不是A的子结构。(约定空树不是任意一个树的子结构)

B是A的子结构，即 A中有出现和B相同的结构和节点值。

例如:给定的树 A:

3 / \ 4 5 / \ 1 2 给定的树 B:

4 / 1 返回 true，因为 B 与 A 的一个子树拥有相同的结构和节点值。

示例：

```
1  输入：A = [1,2,3], B = [3,1]
2  输出：false
```

## 662. 二叉树最大宽度

给定一个二叉树，编写一个函数来获取这个树的最大宽度。树的宽度是所有层中的最大宽度。这个二叉树与满二叉树 (full binary tree) 结构相同，但一些节点为空。

每一层的宽度被定义为两个端点（该层最左和最右的非空节点，两端点间的 null 节点也计入长度）之间的长度。

示例：

```
1  输入：
2
3      1
4     / \
5    3   2
6   / \   \
7  5  3   9
8
9  输出：4
10 解释：最大值出现在树的第 3 层，宽度为 4 (5,3,null,9)。
```

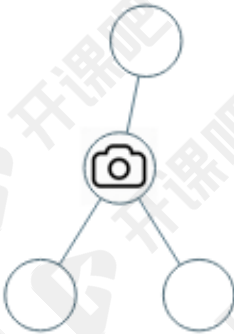
# 968. 监控二叉树

给定一个二叉树，我们在树的节点上安装摄像头。

节点上的每个摄影头都可以监视其父对象、自身及其直接子对象。

计算监控树的所有节点所需的最小摄像头数量。

示例：



1

输入：[0,0,null,0,0]

2

输出：1

3

解释：如图所示，一台摄像头足以监控所有节点