

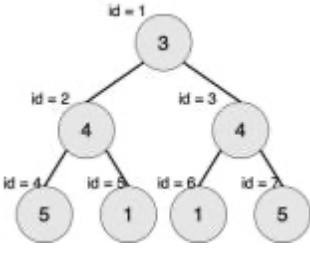
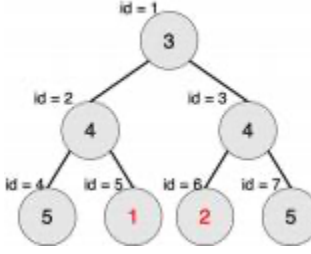
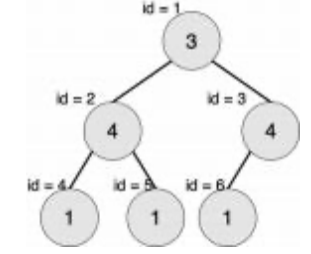
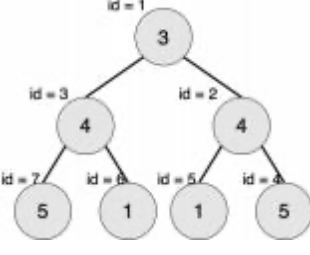
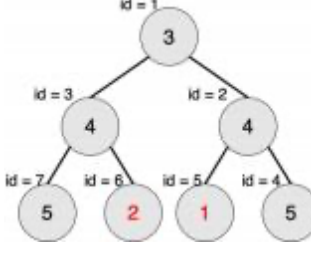
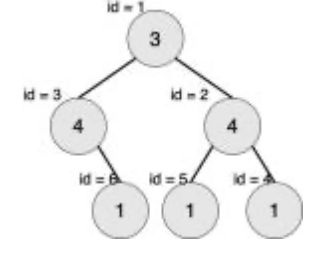
对称二叉树

【问题描述】

一棵有点权的有根树如果满足以下条件，则被轩轩称为**对称二叉树**：

1. 二叉树；
2. 将这棵树**所有**节点的左右子树交换，新树和原树对应位置的结构相同且点权相等。

下图中节点内的数字为权值，节点外的 id 表示节点编号。

	对称二叉树	非对称二叉树 (权值不对称)	非对称二叉树 (结构不对称)
原树			
所有节点的左右子树交换后			

现在给出一棵二叉树，希望你找出它的一棵子树，**该子树为对称二叉树，且节点数最多**。请输出这棵子树的节点数。

注意：只有树根的树也是对称二叉树。本题中约定，以节点 T 为子树根的一棵“子树”指的是：节点 T 和它的**全部**后代节点构成的二叉树。

【输入格式】

第一行一个正整数 n，表示给定的树的节点的数目，规定节点编号 $1 \sim n$ ，其中节点 1 是树根。

第二行 n 个正整数，用一个空格分隔，第 i 个正整数 v_i 代表节点 i 的权值。

接下来 n 行，每行两个正整数 l_i, T_i ，分别表示节点 i 的左右孩子的编号。如果不存在左 / 右孩子，则以 -1 表示。两个数之间用一个空格隔开。

【输出格式】

输出文件共一行，包含一个整数，表示给定的树的最大对称二叉子树的节点数。

【数据规模与约定】

共 25 个测试点。

$v_i \leq 1000$ 。

测试点 $1 \sim 3$ ， $n \leq 10$ ，保证根结点的左子树的所有节点都没有右孩子，根结点的右子树的所有节点都没有左孩子。

测试点 $4 \sim 8$ ， $n \leq 10$ 。

测试点 $9 \sim 12$ ， $n \leq 105$ ，保证输入是一棵“满二叉树”。

测试点 $13 \sim 16$ ， $n \leq 105$ ，保证输入是一棵“完全二叉树”。

测试点 17~20 , $n \leq 105$, 保证输入的树的点权均为 1。

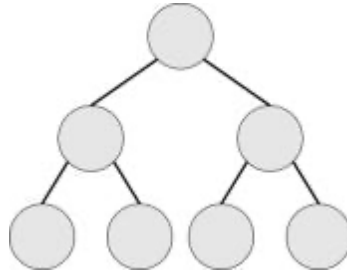
测试点 21~25 , $n \leq 106$ 。

本题约定:

层次: 节点的层次从根开始定义起, 根为第一层, 根的孩子为第二层。树中任一节点的层次等于其父亲节点的层次加 1。

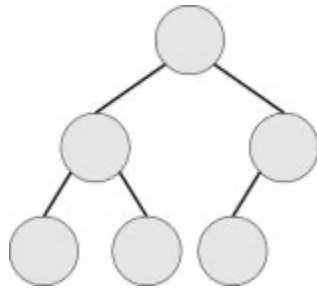
树的深度: 树中节点的最大层次称为树的深度。

满二叉树: 设二叉树的深度为 h , 且二叉树有 $2^h - 1$ 个节点, 这就是满二叉树。

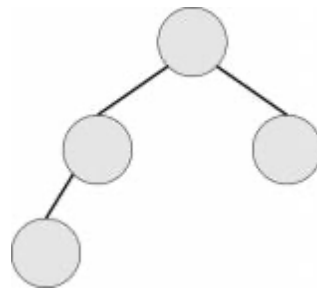


满二叉树 (深度为 3)

完全二叉树: 设二叉树的深度为 h , 除第 h 层外, 其它各层的结点数都达到最大个数, 第 h 层所有的结点都连续集中在最左边, 这就是完全二叉树。



完全二叉树 (深度为 3)



完全二叉树 (深度为 3)