树的深度(depth)

【题目描述】

有一个n个结点的二叉树。给出每个结点的两个子结点编号（均不超过n），建立一棵二叉树（根节点的编号为1），如果是叶子结点，则输入0 0。

建好这棵二叉树之后，请求出它的深度。二叉树的深度是指从根节点到叶子结点时，最多经过了几层。

【输入格式】

第一行一个整数n，表示结点数。

之后n行，第i行两个整数l、r，分别表示结点i的左右子结点编号。若 l = 0 则表示无左子结点，r = 0同理。

【输出格式】

一个整数，表示最大结点深度。

【输入输出样例】

|  |  |
| --- | --- |
| 样例输入 | 样例输出 |
| 7  2 7  3 6  4 5  0 0  0 0  0 0  0 0 | 4 |

【数据规模】

n ≤ 104, 0 ≤ l, r ≤ n

树转二叉树

【题目描述】

树转二叉树，遵循的是左孩子、右兄弟规则，现在给你一棵树，将其转换成二叉树，并输出该二叉树的先序遍历和中序遍历。

【输入格式】

第一行一个整数n，表示结点数。

之后n行，第i行两个整数x、y，示结点x是节点y的父亲。根节点的父亲为 0。

【输出格式】

共两行，每行n个数字，用空格隔开。

第一行是这个二叉树的前序遍历。

第二行是这个二叉树的中序遍历。

【输入输出样例】

|  |  |
| --- | --- |
| 样例输入 | 样例输出 |
| 4  0 2  2 1  2 3  2 4 | 2 1 3 4  1 3 4 2 |

【数据规模】

n ≤ 103

合并果子

【题目描述】

在一个果园里，多多已经将所有的果子打了下来，而且按果子的不同种类分成了不同的堆。多多决定把所有的果子合成一堆。

每一次合并，多多可以把两堆果子合并到一起，消耗的体力等于两堆果子的重量之和。可以看出，所有的果子经过n-1次合并之后， 就只剩下一堆了。多多在合并果子时总共消耗的体力等于每次合并所耗体力之和。

因为还要花大力气把这些果子搬回家，所以多多在合并果子时要尽可能地节省体力。假定每个果子重量都为 1 ，并且已知果子的种类 数和每种果子的数目，你的任务是设计出合并的次序方案，使多多耗费的体力最少，并输出这个最小的体力耗费值。

例如有3种果子，数目依次为1，2，9。可以先将1、2堆合并，新堆数目为3，耗费体力为3。接着，将新堆与原先的第三堆合并，又得到新的堆，数目为12，耗费体力为12。所以多多总共耗费体力 = 3 + 12 = 15。可以证明15为最小的体力耗费值。

【输入格式】

共两行。

第一行是一个整数 n (1≤ n ≤10000)，表示果子的种类数。

第二行包含n个整数，用空格分隔，第i个整数ai(1≤ ai ≤20000)是第i种果子的数目。

【输出格式】

一个整数，也就是最小的体力耗费值。输入数据保证这个值小于231。

【输入输出样例】

|  |  |
| --- | --- |
| 样例输入 | 样例输出 |
| 3  1 2 9 | 15 |

【数据规模】

对于30%的数据，保证有n ≤ 1000；

对于50%的数据，保证有n ≤ 5000；

对于全部的数据，保证有n ≤ 10000。

求重儿子

【题目描述】

给定n个结点的一棵树，求每个结点的重儿子，其中节点1为根结点

【输入格式】

第一行包含一个正整数n，表示树的结点数。

接下来n-1行每行包含两个整数x,y，表示点x和点y之间连有一条边（保证无环且连通）。

【输出格式】

输出共n行，每行包含若干个正整数，用空格隔开。

第i行中先输出一个正整数表示结点i的重儿子个数，然后按编号从小到大的顺序依次输出这些重儿子。

【输入输出样例】

|  |  |
| --- | --- |
| 样例输入 | 样例输出 |
| 5  1 2  1 3  2 4  2 5 | 1 2  2 4 5  0  0  0 |

【数据规模】

n ≤ 105

会议

【题目描述】

有一个村庄居住着n个村民，有n-1条路径使得这n个村民的家联通，每条路径的长度都为1。现在村长希望在某个村民家中召开一场会议，村长希望所有村民到会议地点的距离之和最小，那么村长应该要把会议地点设置在哪个村民的家中，并且这个距离总和最小是多少？若有多个节点都满足条件，则选择节点编号最小的那个点。

【输入格式】

第一行，一个数n，表示有n个村民。

接下来n-1行，每行两个数字a和b，表示村民a的家和村民b的家之间存在一条路径。

【输出格式】

一行输出两个数字x和y。

x表示村长将会在哪个村民家中举办会议。

y表示距离之和的最小值。

【输入输出样例】

|  |  |
| --- | --- |
| 样例输入 | 样例输出 |
| 4  1 2  2 3  3 4 | 2 4 |

【数据规模】

对于70%数据n ≤ 103。

对于100%数据n ≤ 5×104。

拜访农场

【题目描述】

Farmer John 计划建造N个农场，用N-1条道路连接，构成一棵树（也就是说，所有农场之间都互相可以到达，并且没有环）。每个农场有一头奶牛，品种为更赛牛或荷斯坦牛之一。

Farmer John 的M个朋友经常前来拜访他。在朋友i拜访之时，Farmer John 会与他的朋友沿着从农场Ai到农场bi之间的唯一路径行走（可能有Ai = Bi）。除此之外，他们还可以品尝他们经过的路径上任意一头奶牛的牛奶。由于 Farmer John 的朋友们大多数也是农场主，他们对牛奶有着极强的偏好。他的有些朋友只喝更赛牛的牛奶，其余的只喝荷斯坦牛的牛奶。任何 Farmer John 的朋友只有在他们访问时能喝到他们偏好的牛奶才会高兴。

请求出每个朋友在拜访过后是否会高兴。

【输入格式】

输入的第一行包含两个整数N和M。

第二行包含一个长为N的字符串。如果第i个农场中的奶牛是更赛牛，则字符串中第i个字符为G，如果第i个农场中的奶牛是荷斯坦牛则为H。

以下N-1行，每行包含两个不同的整数X和Y（1 ≤ X,Y ≤ N），表示农场X与Y之间有一条道路。

以下M行，每行包含整数Ai，Bi，以及一个字符Ci。Ai和Bi表示朋友i拜访时行走的路径的端点，Ci是G或H之一，表示第i个朋友喜欢更赛牛的牛奶或是荷斯坦牛的牛奶。

【输出格式】

为M的二进制字符串。如果第i个朋友会感到高兴，则字符串的第i个字符为1，否则为0。

【输入输出样例】

|  |  |
| --- | --- |
| 样例输入 | 样例输出 |
| 5 5  HHGHG  1 2  2 3  2 4  1 5  1 4 H  1 4 G  1 3 G  1 3 H  5 5 H | 10110 |

【样例说明】

在这里，从农场1到农场4的路径包括农场1、2和4。所有这些农场里都是荷斯坦牛，所以第一个朋友会感到满意，而第二个朋友不会。

【数据规模】

测试点 1 样例。

测试点 2 - 5 满足 N ≤ 103，M ≤ 2×103。

对于 100% 的数据，1 ≤ N ≤ 105，1 ≤ M ≤ 105。

核心城市

【题目描述】

X国有n座城市n-1条长度为1的道路，每条道路连接两座城市，且任意两座城市都能通过若干条道路相互到达，显然，城市和道路形成了一棵树。

X 国国王决定将k座城市钦定为X国的核心城市，这k座城市需满足以下两个条件：

1. 这k座城市可以通过道路，在不经过其他非核心城市的情况下两两相互到达。

2. 定义某个非核心城市与这k座核心城市的距离为，这座城市与k座核心城市的距离的最小值。那么所有非核心城市中，与核心城市的距离最大的城市，其与核心城市的距离最小。你需要求出这个最小值。

【输入格式】

第一行2个正整数n,k。

接下来n-1行，每行2个正整数u,v，表示第u座城市与第v座城市之间有一条长度为1的道路。

【输出格式】

一行一个整数，表示答案。

【输入输出样例】

|  |  |
| --- | --- |
| 样例输入 | 样例输出 |
| 6 3  1 2  2 3  2 4  1 5  5 6 | 1 |

【数据规模】

1 ≤ k < n ≤ 10 ^ 5。

1 ≤ u,v ≤ n, u ≠ v，保证城市与道路形成一棵树。