

第三章 树

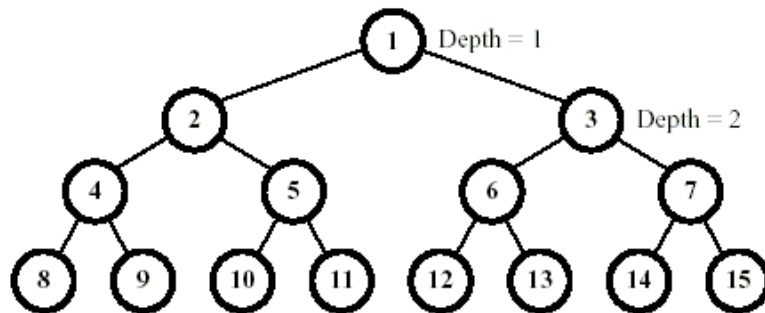
第二节 二叉树

【上机练习】

1、小球(drop)

【问题描述】

许多的小球一个一个的从一棵满二叉树上掉下来组成 FBT (Full Binary Tree, 满二叉树), 每一时间, 一个正在下降的球第一个访问的是非叶子节点。然后继续下降时, 或者走右子树, 或者走左子树, 直到访问到叶子节点。决定球运动方向的是每个节点的布尔值。最初, 所有的节点都是 false, 当访问到一个节点时, 如果这个节点是 false, 则这个球把它变成 true, 然后从左子树走, 继续它的旅程。如果节点是 true, 则球也会改变它为 false, 而接下来从右子树走。满二叉树的标记方法如下图:



因为所有的节点最初为 false, 所以第一个球将会访问节点 1, 节点 2 和节点 4, 转变节点的布尔值后在在节点 8 停止。第二个球将会访问节点 1、3、6, 在节点 12 停止。明显地, 第三个球在它停止之前, 会访问节点 1、2、5, 在节点 10 停止。

现在你的任务是, 给定 FBT 的深度 D, 和 I, 表示第 I 个小球下落, 你可以假定 I 不超过给定的 FBT 的叶子数, 写一个程序求小球停止时的叶子序号。

【输入格式】

输入文件仅一行包含两个用空格隔开的整数 D 和 I。其中 $2 \leq D \leq 20$, $1 \leq I \leq 524288$ 。

【输出格式】

对应输出第 I 个小球下落停止时的叶子序号。

【输入样例】

4 2

【输出样例】

12

2、二叉树遍历(flist)

【问题描述】

树和二叉树基本上都有先序、中序、后序、按层遍历等遍历顺序, 给定中序和其它一种遍历的序列就可以确定一棵二叉树的结构。

假定一棵二叉树一个结点用一个字符描述, 现在给出中序和按层遍历的字符串, 求该树的先序遍历字符串。

【输入格式】

输入文件 flist.in 共两行, 每行是由字母组成的字符串 (一行的每个字符都是唯一的), 分别表示二叉树的中序遍历和按层遍历的序列。

【输出格式】

输出文件 `flist.out` 就一行，表示二叉树的先序序列。

【输入样例】

DBEAC
ABCDE

【输出样例】

ABDEC

3、FBI 树(fbi)

【问题描述】

我们可以把由“0”和“1”组成的字符串分为三类：全“0”串称为 B 串，全“1”串称为 I 串，既含“0”又含“1”的串则称为 F 串。

FBI 树是一种二叉树¹，它的结点类型也包括 F 结点，B 结点和 I 结点三种。由一个长度为 2^N 的“01”串 S 可以构造出一棵 FBI 树 T，递归的构造方法如下：

T 的根结点为 R，其类型与串 S 的类型相同；

若串 S 的长度大于 1，将串 S 从中间分开，分为等长的左右子串 S1 和 S2；由左子串 S1 构造 R 的左子树 T1，由右子串 S2 构造 R 的右子树 T2。

现在给定一个长度为 2^N 的“01”串，请用上述构造方法构造出一棵 FBI 树，并输出它的后序遍历²序列。

【输入文件】

输入文件 `fbi.in` 的第一行是一个整数 N ($0 \leq N \leq 10$)，第二行是一个长度为 2^N 的“01”串。

【输出文件】

输出文件 `fbi.out` 包括一行，这一行只包含一个字符串，即 FBI 树的后序遍历序列。

【样例输入】

3
10001011

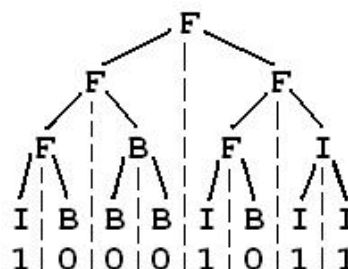
【样例输出】

IBFBBBFIBFIIFF

【数据规模】

对于 40% 的数据， $N \leq 2$ ；

对于 100% 的数据， $N \leq 10$ 。



4、二叉树输出(btout)

【问题描述】

树的凹入表示法主要用于树的屏幕或打印输出，其表示的基本思想是兄弟间等长，一个结点的长度要不小于其子结点的长度。二叉树也可以这样表示，假设叶结点的长度为 1，一个非叶结点的长度等于它的左右子树的长度之和。

一棵二叉树的一个结点用一个字母表示（无重复），输出时从根结点开始：

每行输出若干个结点字符（相同字符的个数等于该结点长度），

如果该结点有左子树就递归输出左子树；

如果该结点有右子树就递归输出右子树。

¹ 二叉树：二叉树是结点的有限集合，这个集合或为空集，或由一个根结点和两棵不相交的二叉树组成。这两棵不相交的二叉树分别称为这个根结点的左子树和右子树。

² 后序遍历：后序遍历是深度优先遍历二叉树的一种方法，它的递归定义是：先后序遍历左子树，再后序遍历右子树，最后访问根。

假定一棵二叉树一个结点用一个字符描述，现在给出先序和中序遍历的字符串，用树的凹入表示法输出该二叉树。

【输入格式】

输入文件 btout.in 共两行，每行是由字母组成的字符串（一行的每个字符都是唯一的），分别表示二叉树的先序遍历和中序遍历的序列。

【输出格式】

输出文件 btout.out 的行数等于该树的结点数，每行的字母相同。

【输入样例】

ABCDEFG

CBDAFEG

【输出样例】

AAAA

BB

C

D

EE

F

G

5、查找二叉树(tree_a)

【问题描述】

已知一棵二叉树用邻接表结构存储，中序查找二叉树中值为 x 的结点，并指出是第几个结点。例：如图二叉树的数据文件的数据格式如下

7

15

5 2 3

12 4 5

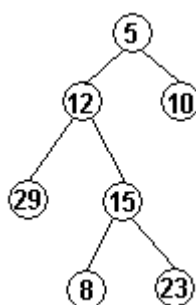
10 0 0

29 0 0

15 6 7

8 0 0

23 0 0



第一行 n 为二叉树的结点个数， $n \leq 100$ ；第二行 x 表示要查找的结点的值；以下第一列数据是各结点的值，第二列数据是左儿子结点编号，第三列数据是右儿子结点编号。

输出一个数即查找的结点编号。

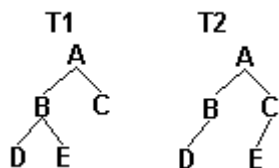
本例的输出样例: 4

6、对称二叉树(tree_c)

【问题描述】

如果二叉树的左右子树的结构是对称的，即两棵子树皆为空，或者皆不空，则称该二叉树是对称的。编程判断给定的二叉树是否对称。

例：如下图中的二叉树 T1 是对称的，T2 是不对称的。



二叉树用顺序结构给出，若读到#则为空，二叉树 $T1=ABCDE$ ， $T2=ABCD\#E$ ，如果二叉树是对称的，输出 “Yes”，反之输出 “No”。

【输入样例】

ABCDE

【输出样例】

Yes

第三节 堆及其应用

【上机练习】

1、合并果子(fruit)

【问题描述】

在一个果园里，多多已经将所有果子打了下来，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。

【输入文件】

输入文件 fruit.in 包括两行，第一行是一个整数 n ($1 \leq n \leq 30000$)，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 a_i ($1 \leq a_i \leq 20000$) 是第 i 种果子的数目。

【输出文件】

输出文件 fruit.out 包括一行，这一行只包含一个整数，也就是最小的体力耗费值。输入数据保证这个值小于 2^{31} 。

【样例 1 输入】

3
1 2 9

【样例 1 输出】

15

【样例 2 输入】

10
3 5 1 7 6 4 2 5 4 1

【样例 2 输出】

120

2、最小函数值(minval)**【问题描述】**

有 n 个函数，分别为 F_1, F_2, \dots, F_n 。定义 $F_i(x) = A_i * x^2 + B_i * x + C_i (x \in \mathbb{N}^*)$ 。给定这些 A_i, B_i 和 C_i ，请求出所有函数的所有函数值中最小的 m 个（如有重复的要输出多个）。

【输入格式】

第一行输入两个正整数 n 和 m 。

以下 n 行每行三个正整数，其中第 i 行的三个数分别位 A_i, B_i 和 C_i 。输入数据保证 $A_i \leq 10, B_i \leq 100, C_i \leq 10\ 000$ 。

【输出格式】

输出将这 n 个函数所有可以生成的函数值排序后的前 m 个元素。

这 m 个数应该输出到一行，用空格隔开。

【输入样例】

3 10
4 5 3
3 4 5
1 7 1

【输出样例】

9 12 12 19 25 29 31 44 45 54

【数据规模】

$n, m \leq 10\ 000$