

# 实验五 树和二叉树

## 一、目的和要求

1. 掌握二叉树的逻辑结构定义和各种存储结构的实现。
2. 熟练运用二叉树的各种存储结构以及各种基本操作。
3. 根据实际问题的需要, 选择二叉树适合的存储结构解决问题。

## 三、实验内容

### (一) 验证性实验

#### 1. 二叉链表的验证

- (1) 在二叉链表类模板中增加函数成员 `CountLeaf()`, 统计二叉树中叶子结点的数目。
- (2) 在二叉链表类模板中增加函数成员 `Revolute()`, 实现二叉树中所有结点的左右子树交换。
- (3) 在二叉链表类模板中增加函数成员 `CountBreadth()`, 统计二叉树的最大宽度(二叉树的最大宽度是指二叉树所有层中结点个数的最大值)。
- (4) 在二叉链表类模板中, 增加函数成员 `NonRecurringInOrder()`, 实现非递归中序遍历二叉树。

#### 2. 线索二叉树的验证

- (1) 在中序线索二叉树类模板中增加函数成员 `ReInOrder()`, 通过从中序序列最后一个结点开始依次找当前结点的前驱来逆中序遍历二叉树。
- (2) 在中序线索二叉树类模板中增加函数成员 `InsertLeftChild(p,e)`, 实现在中序线索二叉树指定结点 `p` 上插入左孩子结点 `e`。
- (3) 在中序线索二叉树类模板中增加函数成员 `PostOrder()`, 实现不用栈后序遍历二叉树。

### (二) 设计性实验

#### 3. 二叉树的顺序存储

参考二叉树的二叉链表类模板, 设计并实现二叉树的顺序存储表示。增加函数成员, 求离两个元素(编号为 `i` 和 `j`) 最近共同祖先。

#### 4. 二叉树的三叉链表表示

参考二叉树的二叉链表类模板, 设计并实现二叉树的三叉链表表示。增加函数成员, 实现非递归先序、后序遍历二叉树。

#### 5. 后序线索二叉树的实现

参考中序线索二叉树的类模板, 设计并实现后序线索二叉树。增加函数成员, 利用线索求指定结点 `p` 在后序序列中的后继结点。

#### 6. 先序线索二叉树的实现

参考中序线索二叉树的类模板, 设计并实现先序线索二叉树。增加函数成员, 利用线索求指定结点 `p` 在先序序列中的后继结点。

### (三) 综合性实验

#### 7. 标记二叉树

### [问题描述]

一棵二叉树，根节点标记为(1,1)，规定：如果一个结点标记为(a,b)，则它的左孩子(如果存在)标记为(a+b,b)，它的右孩子(如果存在)标记为(a,a+b)。现在已知某个结点的标记为(a,b)，求从根节点开始需要经过多少次左分支和多少次右分支才能到达结点(a,b)。

### [输入文件]

输入文件第一行只有一个整数n，表示测试的数据组数。

接下来n行(第2~n+1行)，每行包括二个整数a和b。

### [输出文件]

输出文件有n行，每行包括二个整数，分别表示从根节点开始达结点(a,b)需要进过的左分支数和右分支数。

### [输入样例]

```
2
42 1
3 4
```

### [输出样例]

```
41 1
2 1
```

## 8. 小球下落

### [问题描述]

有一棵高度为h的满二叉树，所有结点从上到下、自左向右编号为：1、2、……、 $2^h-1$ 。在根结点处放一个小球，它会沿着分支往下滑落到叶子结点。除叶子结点外，每个结点上都有一个开关，初始全部关闭，当有小球经过结点时，开关状态会改变。小球到达结点时，如果该结点开关处于关闭状态，则小球往左分支滑落，否则往右分支滑落，直到滑到叶子结点。

### [输入文件]

输入多组数据，每组数据一行两个数，分别表示满二叉树的高度h( $h \leq 20$ )和小球数目n( $n \leq 2^{h-1}$ )。

### [输入文件]

输出第n个小球所滑到的叶子结点编号。

### [输入样例]

```
4 2
3 4
10 1
2 2
8 128
16 1345
```

### [输出样例]

```
12
7
512
3
255
36358
```

## 9. 强盗团伙

### [问题描述]

1920 年的芝加哥，出现了一群强盗。如果两个强盗遇上了，那么他们要么是朋友，要么是敌人。而且有一点是肯定的，就是 A 的朋友的朋友是 A 的朋友；A 的敌人的敌人也是 A 的朋友。两个强盗是同一伙当且仅当他们是朋友。现在给你一些关于强盗们的信息，问你至多有多少个强盗团伙

**[输入数据]**

输入的第一行为 N( $2 \leq N \leq 1000$ )，表示强盗的个数（从 1 编号到 N）。

第二行 M( $1 \leq M \leq 100000$ )，表示信息条数。

以下 M 行，每行可能是 F p q 或是 E p q，分别表示 p 和 q 是朋友，或是敌人。

假设输入不会产生矛盾

**[输出数据]**

输出只有一行，表示最大可能的强盗团伙数。

**[输入样例]**

```
6
4
E 1 4
F 3 5
F 4 6
E 1 2
```

**[输出样例]**

```
3
```

**[测试数据要求]**

输入数据在 gangs.in 文件中；输出数据在 gangs.out 文件中。

## 10. 食物链

**[问题描述]**

动物王国中有三类动物 A,B,C，这三类动物的食物链构成了有趣的环形。A 吃 B， B 吃 C， C 吃 A。

现有 N 个动物，以 1—N 编号。每个动物都是 A,B,C 中的一种，但是我们并不知道它到底是哪一种。

有人用两种说法对这 N 个动物所构成的食物链关系进行描述：

第一种说法是"1 X Y"，表示 X 和 Y 是同类。

第二种说法是"2 X Y"，表示 X 吃 Y。

此人对 N 个动物，用上述两种说法，一句接一句地说出 K 句话，这 K 句话有的是真的，有的是假的。当一句话满足下列三条之一时，这句话就是假话，否则就是真话。

- 1) 当前的话与前面的某些真的话冲突，就是假话；
- 2) 当前的话中 X 或 Y 比 N 大，就是假话；
- 3) 当前的话表示 X 吃 X，就是假话。

你的任务是根据给定的 N ( $1 \leq N \leq 50,000$ ) 和 K 句话 ( $0 \leq K \leq 100,000$ )，输出假话的总数。

**[输入数据]**

输入数据在 foodchain.in 文件中，第一行是两个整数 N 和 K，以一个空格分隔。

以下 K 行每行是三个正整数 D, X, Y，两数之间用一个空格隔开，其中 D 表示说法的种类。

若 D=1，则表示 X 和 Y 是同类。

若 D=2，则表示 X 吃 Y。

**[输出数据]**

输出数据在 foodchain.out 文件中，只有一个整数表示假话的数目。

**[输入样例]**

```
100 7
1 101 1
```

2 1 2

2 2 3

2 3 3

1 1 3

2 3 1

1 5 5

[输出样例]

3