**[并查集学习小节(POJ版)](http://www.cppblog.com/abilitytao/archive/2009/10/18/98899.html)**

什么是并查集呢，我相信大家都已经很熟悉了，在这里我就不再赘述。写这篇文章的主要目的不是新手教学，而是为了借POJ上相关的题目，全面的总结一下并查集问题和它的各个变种。  
  
POJ 1611 The Suspects  
题目大意:有n个学生（标号为0 to n-1),m个学生社团，给出每个社团里所有学生的标号，并假设0号学生患有SARS（社团里只要用一个学生患病，则整个社团里的学生都会被隔离），问最后一共会有多少学生被隔离？  
这是一个最基础的并查集的应用，扫描每一个社团，只要两个学生出现在同一个社团，则将这两个集合合并起来，最后输出0号点所在集合的rank值集合（rank值记录这个集合中的元素个数并用一个flag值跟踪0号元素所在集合标号）即可。  
这是并查集问题的第一种应用：集合合并，判断两点是不是在同一个集合，求某一个集合上的元素个数等。

http://www.cppblog.com/Images/OutliningIndicators/None.gif#include<stdio.h>  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif#define MAX 30000  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint f[MAX];//这里的1001只是一个示意性的数字 代表初始状态下的分支数目  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint r[MAX];  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint flag;  
http://www.cppblog.com/Images/OutliningIndicators/None.gif//由于不知道应该将子树挂到那个集合上面去，故需要一个准则，这里的准则是将子树挂到  
http://www.cppblog.com/Images/OutliningIndicators/None.gif//r值大的集合上面去,初始状态下r数组的值均为一，代表每个分支下只有一个数字  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint find(int n)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    if(f[n]==n)  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        return n;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    else  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        f[n]=find(f[n]);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return f[n];  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}//查找函数，并压缩路径  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint Union(int x,int y)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int a=find(x);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int b=find(y);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    if(a==b)  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        return 0;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    else if(r[a]<=r[b])  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        f[a]=b;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        r[b]+=r[a];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(a==flag)  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            flag=b;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    else  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        f[b]=a;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        r[a]+=r[b];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(b==flag)  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            flag=a;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return 1;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif      
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}//合并函数，如果属于同一分支则返回0，成功合并返回1  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint main()  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int n,m;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i,j;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int num;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int maxnum=0;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    while(scanf("%d%d",&n,&m))  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        flag=0;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        maxnum=0;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        int temp1,temp2;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(n==0&&m==0)  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            break;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        for(i=0;i<n;i++)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            f[i]=i;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            r[i]=1;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        for(j=1;j<=m;j++)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            scanf("%d",&num);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            for(i=0;i<num;i++)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                if(i==0)  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                    scanf("%d",&temp1);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                else  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif                http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                    scanf("%d",&temp2);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                    Union(temp1,temp2);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif                }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        printf("%d\n",r[flag]);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif          
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return 0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif

POJ 2492 A Bug's Life  
个人认为它是初级并查集问题的一个升级。同时这个题让我看到了食物链的影子。。。  
题目的大意是给出n只bug和m次观察到的性行为，并以此为依据判断两只bugs是不是有同性恋行为(gay)。  
比如3只bug  
1 2有性行为  
2 3有性行为  
1 3有性行为  
---->>>>>首先1,2是异性。  
---->>>>>然后2,3是异性。  
可以推出1,3是异性。  
但是1，3有性行为，所以可以判断出有一定有同性恋。  
  
剥离这个题目所赋予的外壳，我们抽出这个问题的本质：并查集！  
其实，这里最重要的是去维护每一个点到集合顶点的偏移量。（注意：下面生造了一个词 所谓集合元素 比如说f[i]=i,那么i就是集合元素，集合偏移量就是集合元素的偏移量）  
  
初始状态下，应该是   
i号点挂在i号集合下面  
我们考虑一般情况：假设合并的过程已经进行了一部分 ，这样每一个集合下面都有元素，且各自对于顶点的偏移量都算出来了；  
现在在a集合中的元素x和b集合中的元素y进行合并。此时有两中情况改变偏移量；  
1.首先是集合的合并，如果要将a,b集合合并，又要保证x,y数字的kind不相同，比如说把b集合挂到a集合下面去。  
代表集合的那个元素，他的偏移量永远是0，所以b要改变偏移量，使得b里面的y在进行变换后要和x相异。  
如果 kind[x]=0;kind[y]=0;那么y对应的那个代表集合的元素的偏移量必须变成1，因为只有这样才能使得合并后，x,y有不同的kind;  
如果 kind[x]=0,kind[y]=1;y对应代表集合的元素偏移量是0，所以对应集合偏移量还是0；  
类推   kind[x]=1,kind[y]=0,同上,0；  
           kind[x]=1,kind[y]=1,y集合偏移量应该变为1；  
综上 可以得到一个同或的关系。  
用等式 kind[a]=(kind[x]+kind[y]+1)%2;恰好满足要求.  
2.然后是压缩路径时候的偏移量改变  
个人认为，这个主要是解决集合合并时候产生的“残余问题”，因为在合并集合的时候只是考虑了集合的偏移量，至于它下面的元素一概不管。一个压缩路径既分离了父子元素的偏移量，又使得子元素直接指向集合元素。  
  
总而言之，并查集的操作就是不断地维护者各个集合中，每个元素身上对集合元素的偏移关系。从而确定他们是否具有同性恋。  
在这个题中，假设是不存在同性恋的，所以只有找到矛盾才输出 有同性恋。

http://www.cppblog.com/Images/OutliningIndicators/None.gif#include<iostream>  
http://www.cppblog.com/Images/OutliningIndicators/None.gif#include<cstdio>  
http://www.cppblog.com/Images/OutliningIndicators/None.gifusing namespace std;  
http://www.cppblog.com/Images/OutliningIndicators/None.gif#define MAX 2001  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint f[MAX];  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint kind[MAX];  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint n,m;  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint testcase;  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifvoid init()  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    for(i=1;i<=n;i++)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{   
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        f[i]=i;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        kind[i]=0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint Find(int n)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif      
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    if(f[n]==n)  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        return n;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int t=Find(f[n]);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    kind[n]=(kind[n]+kind[f[n]])%2;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    f[n]=t;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return f[n];  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint  Union(int x,int y)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif      
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int a=Find(x);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int b=Find(y);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    if(a==b)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(kind[x]==kind[y])  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            return 1;//1代表有同性恋情况  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    else   
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        f[a]=b;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        kind[a]=(kind[x]+kind[y]+1)%2;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return 0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint main()  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    scanf("%d",&testcase);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i,j;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int a,b;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int flag;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    for(i=1;i<=testcase;i++)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        flag=0;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        scanf("%d%d",&n,&m);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        init();  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        for(j=1;j<=m;j++)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            scanf("%d%d",&a,&b);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            if(Union(a,b))  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                flag=1;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(flag==1)  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            printf("Scenario #%d:\nSuspicious bugs found!\n\n",i);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        else   
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            printf("Scenario #%d:\nNo suspicious bugs found!\n\n",i);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return 0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif

POJ 1182 食物链   
中文题，让你输出假话的个数。其实这道题是上一道题的扩展，如果把上一道题也想成是食物链的话，就是1吃2，2吃1.  
而这里是三个动物，所以同样是维护一个偏移量，只不过多了一位罢了。  
程序的过程实质上就是在维护并查集，判断是否是假话是在维护的过程中进行的，只能算是附属品吧。

http://www.cppblog.com/Images/OutliningIndicators/None.gif#include<iostream>  
http://www.cppblog.com/Images/OutliningIndicators/None.gifusing namespace std;  
http://www.cppblog.com/Images/OutliningIndicators/None.gif#define MAX 50005  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint f[MAX];  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint kind[MAX];  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint n,m;  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifvoid init()  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    for(i=1;i<=n;i++)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{   
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        f[i]=i;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        kind[i]=0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint Find(int n)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif      
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    if(f[n]==n)  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        return n;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int t=Find(f[n]);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    kind[n]=(kind[n]+kind[f[n]])%3;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    f[n]=t;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return f[n];  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gifbool  Union(int x,int y,int c)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    if(x>n||y>n)  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        return 1;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int a=Find(x);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int b=Find(y);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    if(c==1)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(a==b)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            if(kind[x]!=kind[y])  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                return true;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        else if(a!=b)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            f[b]=a;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            kind[b]=(kind[x]-kind[y]+3)%3;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    else  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(x==y)  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            return true;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(a==b)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            if((kind[x]+1)%3!=kind[y])  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                return true;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        else if(a!=b)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            f[b]=a;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            kind[b]=(kind[x]-kind[y]+4)%3;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return false;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint main()  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i,j;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int a,b,c;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int sum=0;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    scanf("%d%d",&n,&m);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    init();  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    for(i=1;i<=m;i++)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        scanf("%d%d%d",&c,&a,&b);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(Union(a,b,c))  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            sum++;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    printf("%d\n",sum);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return 0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif

 这里将两个集合并起来并将所挂集合偏移量指向：  
kind[b]=(kind[x]-kind[y]+4)%3;  
想想上一题是不是也很类似呢  
其实上一题的公式也可以改成  
kind[b]=(kind[x]-kind[y]+3)%2;   
不管是几个动物循环，都能得到类似的结论，所以以后碰到4，5，6，7。。。个动物的食物链，你应该也会做了吧？^\_^  
  
POJ 1988 Cube Stacking  
这道题更有意思了，说它开辟了并查集问题的新局面并不为过；上面2道题，研究的主要是到集合元素的偏移量，而这道题要求的是一个“逻辑上”到达集合元素的距离！集合合并的时候同样只修改被挂集合元素的距离值，残余部分留给压缩路径来处理.  
如果理解了上面的问题，这个问题就很好理解了。

http://www.cppblog.com/Images/OutliningIndicators/None.gif#include<iostream>  
http://www.cppblog.com/Images/OutliningIndicators/None.gif#include<algorithm>  
http://www.cppblog.com/Images/OutliningIndicators/None.gif#include<cmath>  
http://www.cppblog.com/Images/OutliningIndicators/None.gifusing namespace std;  
http://www.cppblog.com/Images/OutliningIndicators/None.gif#define MAX 30000  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint f[MAX+1];  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint r[MAX+1];  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint above[MAX+1];  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifvoid init()  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    for(i=1;i<=MAX;i++)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        above[i]=0;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        f[i]=i;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        r[i]=1;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint realfather;  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint find(int n)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int t;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    if(f[n]==n)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        realfather=n;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        return n;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    else  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        t=find(f[n]);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(f[n]!=realfather)  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            above[n]+=(above[f[n]]);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        f[n]=t;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return f[n];  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}//查找函数，并压缩路径  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifvoid Union(int x,int y)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int a=find(x);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int b=find(y);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    f[b]=a;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    above[b]+=r[a];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    r[a]+=r[b];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif      
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif      
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}//合并函数，如果属于同一分支则返回0，成功合并返回1  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint main()  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int p;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    init();  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    char order;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int a,b;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    scanf("%d",&p);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    for(i=1;i<=p;i++)  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        cin.ignore();  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        scanf("%c",&order);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(order=='M')  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            scanf("%d%d",&a,&b);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            Union(a,b);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        else if(order=='C')  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            scanf("%d",&a);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            printf("%d\n",r[find(a)]-above[a]-1);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return 0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif

银河英雄传说 NOI 2002  
说道并查集，还有一道非常经典的题目 还有那个“著名”的杨威利元帅，呵呵。这题附上原题，有了上面的讲解，相信你能很快找到解法^\_^