并查集 存在树形关系的实际问题，解决时采用并查集会很方便

数据结构

用一个int father[1000..00]来用作并查集，根据题目中的索引关系来确定节点之间的树形关系。

再用一个data数组来存储数据结构体。

算法：

初始化father数组

for{int i=0;i<n;i++){  
 father[i]=i;

}

找父节点

void findfather(int x){

int a=x;  
 *while(x!=father(x)){  
 x = father[x];*

*}*

*while (a!=father[a]) //x到根节点上的节点进行路径压缩*

*{*

*int z = a;*

*a = father[a];*

*father[z] = x;*

*}*

*return x;*

*}*

*合并树*

*void Union(int a,int b){*

*int a1=findfather(a);*

*Int b1=findfather(b);*

*If(a1>b1){ //这里可以把树的根取成树中所有节点中最小的*

*father[a1]=b1; //把a1所在的树插入到b1上*

*}else if(b1>a1){*

*father[b1]=a1; //把b1所在的树插入到a1上*

*}*

*}*

*并查集的使用 找一个节点的根，如果根相同表示这两个节点在同一棵树上。*

*读入数据 构建并查集*

*for (int i = 1; i <= n; i++)*

*{*

*cin >> m;*

*int last;*

*cin >> last;*

*for (int j = 1; j < m; j++)*

*{*

*int dex;*

*cin >> dex;*

*Union(last, dex);*

*last = dex;*

*}*

*}*