

## 并查集

- 1.维护无向图的连通性，支持判断两个点是否在一个联通块里
- 2.判断增加一条边是否会产生环，用在求解最小生成树的kruskal算法中

```
#include <bits/stdc++.h>
#define maxn 1005
typedef long long ll;
using namespace std;
//-----AC(^-^)AC-----\\

struct Node
{
    int pre;//父节点
    int deep;//深度
}node[maxn];
int get_pre(int x)//查找父节点
{
    if(node[x].pre==x)
        return node[x].pre;
    return node[x].pre=get_pre(node[x].pre);//路径压缩
}
void unite(int a,int b)//合并两个集合
{
    a=get_pre(a);
    b=get_pre(b);
    if(node[a].deep>node[b].deep)
        node[b].pre=a;
    else{
        node[a].pre=b;
        if(node[a].deep==node[b].deep)
            node[b].deep++;
    }
}
void build();//初始化
{
    for(int i=1;i<=maxn;i++)
    {
        node[i].pre=i;
        node[i].deep=0;
    }
    return ;
}
int main()
{
    int n,m;
    while(~scanf("%d",&n)&&n)
    {
        build();
```

```
scanf("%d",&m);
for(int i=0;i<m;i++){
    int x,y;
    scanf("%d%d",&x,&y);
    unite(x,y);
}
int ans=0;
for(int i=1;i<=n;i++)
{
    if(node[i].pre==i)
        ans++;
}
printf("%d\n",ans-1);
}
return 0;
}
```