# 1213 How Many Tables

**Problem Description**

Today is Ignatius' birthday. He invites a lot of friends. Now it's dinner time. Ignatius wants to know how many tables he needs at least. You have to notice that not all the friends know each other, and all the friends do not want to stay with strangers.  
  
One important rule for this problem is that if I tell you A knows B, and B knows C, that means A, B, C know each other, so they can stay in one table.  
  
For example: If I tell you A knows B, B knows C, and D knows E, so A, B, C can stay in one table, and D, E have to stay in the other one. So Ignatius needs 2 tables at least.

**Input**

The input starts with an integer T(1<=T<=25) which indicate the number of test cases. Then T test cases follow. Each test case starts with two integers N and M(1<=N,M<=1000). N indicates the number of friends, the friends are marked from 1 to N. Then M lines follow. Each line consists of two integers A and B(A!=B), that means friend A and friend B know each other. There will be a blank line between two cases.

**Output**

For each test case, just output how many tables Ignatius needs at least. Do NOT print any blanks.

**Sample Input**

2

5 3

1 2

2 3

4 5

5 1

2 5

**Sample Output**

2

4

代码清单：

#include<iostream>

#include<cstring>

#include<cstdio>

#include<cstdlib>

using namespace std;

/\*

\* x代表例题中的人，father[x]中所存的数代表这一集合中所有人都与一个人有亲戚关系

\* 相当于例题中第一个集合所有的元素都与第一个元素有亲戚关系

\* 搜索时只要找元素所指向的father[x]=x的元素(即父元素)

\* 然后遍历一遍，father[x]=x占一张桌子，计数输出即可

\*/

int father[1010];

int find(int x)

{

if(father[x]!=x)

father[x]=find(father[x]);

return father[x];

}

int main()

{

int a,b,t,m,n,i,count;

scanf("%d",&t);

while(t--)

{

scanf("%d%d",&n,&m);

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

father[i]=i;

for(i=1;i<=m;i++)

{

scanf("%d%d",&a,&b);

a=find(a);

b=find(b);

father[a]=b;//题目已给a!=b

}

count = 0;

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

if(father[i]==i)

count++;

printf("%d\n",count);

}

return 0;

}

# 1232 畅通工程

**Problem Description**

某省调查城镇交通状况，得到现有城镇道路统计表，表中列出了每条道路直接连通的城镇。省政府“畅通工程”的目标是使全省任何两个城镇间都可以实现交通（但不一定有直接的道路相连，只要互相间接通过道路可达即可）。问最少还需要建设多少条道路？

**Input**

测试输入包含若干测试用例。每个测试用例的第1行给出两个正整数，分别是城镇数目N ( < 1000 )和道路数目M；随后的M行对应M条道路，每行给出一对正整数，分别是该条道路直接连通的两个城镇的编号。为简单起见，城镇从1到N编号。   
注意:两个城市之间可以有多条道路相通,也就是说  
3 3  
1 2  
1 2  
2 1  
这种输入也是合法的  
当N为0时，输入结束，该用例不被处理。

**Output**

对每个测试用例，在1行里输出最少还需要建设的道路数目。

**Sample Input**

4 2

1 3

4 3

3 3

1 2

1 3

2 3

5 2

1 2

3 5

999 0

0

**Sample Output**

1

0

2

998

***Hint***

Hint

Huge input, scanf is recommended.

代码清单：

#include<iostream>

#include<cstring>

#include<cstdio>

#include<cstdlib>

using namespace std;

int father[1010];

int find(int x)

{

if(father[x]!=x)

father[x]=find(father[x]);

return father[x];

}

int main()

{

int a,b,m,n,i,count;

while(scanf("%d",&n),n)

{

scanf("%d",&m);

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

father[i]=i;

for(i=1;i<=m;i++)

{

scanf("%d%d",&a,&b);

a=find(a);

b=find(b);

if(a!=b)

father[a]=b;

}

count = 0;

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

if(father[i]==i)

count++;

printf("%d\n",count-1);

}

return 0;

}