

NOIP 模拟赛

day2

题目名称	受欢迎的牛	封锁	开门问题
题目类型	传统型	传统型	传统型
可执行文件名	cow.exe	blockade.exe	door.exe
输入文件名	cow.in	blockade.in	door.in
输出文件名	cow.out	blockade.out	door.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
测试点/包数目	10	10	10
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	cow.cpp	blockade.cpp	door.cpp
对于 C 语言	cow.c	blockade.c	door.c
对于 Pascal 语言	cow.pas	blockade.pas	door.pas

编译选项

对于 C++ 语言	-O2 -std=c++14 -Wl,--stack=536870912
对于 C 语言	-O2 -std=c14 -Wl,--stack=536870912
对于 Pascal 语言	-O2

受欢迎的牛 (cow)

【题目描述】

每头奶牛都梦想成为牛棚里的明星。被所有奶牛喜欢的奶牛就是一头明星奶牛。所有奶牛都是自恋狂，每头奶牛总是喜欢自己的。奶牛之间的“喜欢”是可以传递的——如果 A 喜欢 B，B 喜欢 C，那么 A 也喜欢 C。牛栏里共有 N 头奶牛，给定一些奶牛之间的爱慕关系，请你算出有多少头奶牛可以当明星。

【输入格式】

从文件 *cow.in* 中读入数据。

第一行：两个用空格分开的整数：N 和 M

第二行到第 M + 1 行：每行两个用空格分开的整数：A 和 B，表示 A 喜欢 B

【输出格式】

输出到文件 *cow.out* 中。

第一行：单独一个整数，表示明星奶牛的数量

【样例 1 输入】

```
3 3
1 2
2 1
2 3
```

【样例 1 输出】

```
1
```

【子任务】

10% 的数据 $N \leq 20, M \leq 50$

30% 的数据 $N \leq 1000, M \leq 20000$

70% 的数据 $N \leq 5000, M \leq 50000$

100% 的数据 $N \leq 10000, M \leq 50000$

封锁 (blockade)

【题目描述】

在 Byteotia 有 n 个城镇。一些城镇之间由无向边连接。在城镇外没有十字路口，尽管可能有桥，隧道或者高架公路（反正不考虑这些）。每两个城镇之间至多只有一条直接连接的道路。人们可以从任意一个城镇直接或间接到达另一个城镇。每个城镇都有一个公民，他们被孤独所困扰。事实证明，每个公民都想拜访其他所有公民一次（在主人所在的城镇）。所以，一共会有 $n * (n-1)$ 次拜访。

不幸的是，一个程序员总罢工正在进行中，那些程序员迫切要求购买某个软件。

作为抗议行动，程序员们计划封锁一些城镇，阻止人们进入，离开或者路过那里。

正如我们所说，他们正在讨论选择哪些城镇会导致最严重的后果。

编写一个程序：

读入 Byteotia 的道路系统，对于每个被决定的城镇，如果它被封锁，有多少访问不会发生，输出结果。

【输入格式】

从文件 *blockade.in* 中读入数据。

第一行两个整数 n, m , 表示城镇数和边数。

接下来 m 行，描述每条边的两个端点。

【输出格式】

输出到文件 *blockade.out* 中。

输出 n 行，每行一个整数，表示封锁第 i 个城市后有多少对访问不会发生。

【样例 1 输入】

```
5 5
1 2
2 3
1 3
3 4
4 5
```

【样例 1 输出】

8
8
16
14
8

【样例 1 解释】

对于 1 号点，有 8 对关系无法达成：(1,2),(2,1),(1,3),(3,1),(1,4),(4,1),(1,5),(5,1)。

【子任务】

对于 40% 数据， $n \leq 1000, m \leq 5000$ 。对于 100% 数据， $n \leq 100000, m \leq 500000$

开门问题 (door)

【题目描述】

一家旅馆的 n 间不同的房间里困住了 n 个人，其中有些房间是被锁住了，有些房间是打开的，但是只有在所有房间同时打开的情况下，被困人员才能逃离。现在有 m 个开关，每个开关控制着一些房间的门，开关的作用是使得这些房间原来开着的关上，关上的打开，但每个门都被两个开关控制。

【输入格式】

从文件 *door.in* 中读入数据。

输入包含 T 组数据。

对于每组数据：

第一行，有两个正整数 n 和 m ， n 表示房间的数量， m 表示开关的数量。第二行 n 个数，表示每个房间的状态，0 表示房间是锁住的，1 表示房间是开着的。再接下来 m 行，每行第一个数 x 表示第 i 把锁控制的房间数，再接着有 x 个数，分别表示所控制的房间编号 ($1 \sim n$)。

【输出格式】

输出到文件 *door.out* 中。

输出 T 行。

对于每组数据，如果房间都能被打开则输出 “YES”，否则输出 “NO”

【样例 1 输入】

```
3
3 3
1 0 1
2 1 3
2 1 2
2 2 3
3 3
1 0 1
3 1 2 3
1 2
2 1 3
3 3
```

```
1 0 1
3 1 2 3
2 1 2
1 3
```

【样例 1 输出】

```
NO
YES
NO
```

【子任务】

对于 100% 数据, $T \leq 5, n, m \leq 100000$.

对于 30% 数据, $n, m \leq 20$