

NOI模拟赛1

saffah

2015 年 5 月 29 日

竞赛时间：2333年3月33日3:33~33:33

中文题目名称	研究小组	道路修建	命名系统
英文题目名称	study	road	name
输入文件名	study.in	road.in	name.in
输出文件名	study.out	road.out	name.out
每个测试点时限	1秒	1秒	3秒
测试点数目	10	20	20
每个测试点分值	10	5	5
内存限制	256MB	256MB	256MB
是否有部分分	否	否	否
题目类型	传统	传统	传统

阿斯大法好

1 研究小组

1.1 题目描述

小P和小Z是一对非常好的朋友，他们所在的学校一共有 n 个人，今天学校要把这些人分成若干个小组做研究。

我们一共有 m 个研究课题，那么根据每个人所选的课题就能把所有自然地分成 m 组。每个人对每个课题有一个讨厌程度，具体地，第 i 个人对第 j 个课题的讨厌程度为 $a_{i,j}$ 。

现在你来扮演学校的角色！你要把这 n 个人分到 m 个课题中去。我们定义某个课题的负能量值，等于参与这个课题的所有人对这个课题的讨厌程度的最大值；如果没有人参与这个课题，那么负能量值为0。现在请你最小化这 m 个课题产生的负能量值之和！

1.2 输入格式

输入文件为study.in。

第一行有两个整数 n, m 。

接下来 n 行，每行 m 个整数，其中第 i 行的第 j 个数是 $a_{i,j}$ 。

1.3 输出格式

输出文件为study.out。

只输出一个整数，表示 m 个课题产生的负能量值之和的最小值。

1.4 样例输入

```
3 3
1 2 100
100 4 5
1 100 3
```

1.5 样例输出

```
5
```

1.6 样例解释

让第1和第3个人参加课题1，让第2个人参加课题2，没人参加课题3。
总计的负能量是 $1 + 4 + 0 = 5$ ，没有更优的方案。

1.7 数据规模与约定

除样例外，所有的 $a_{i,j}$ 是在 $(0, 10^9]$ 范围内均匀随机生成的。

测试点编号	n	m
1	1000000	1
2	500000	2
3	200000	2
4	100000	3
5	50000	3
6	20000	4
7	10000	4
8	5000	4
9	2000	4
10	1000	4

2 道路修建

2.1 题目描述

小P和小Z是一对非常好的朋友，今天他们在玩一个模拟建设类游戏。
游戏中共有 n 个城市，通过 m 条双向道路连接。第 i 条道路连接了城市 a_i 和 b_i 。

不幸的是，在一次巨大的灾难以后，这 m 条道路全部损坏了。修复第 i 条道路需要 c_i 天。把这些道路全部修复的代价可能太大，小P和小Z只希望某 k 个城市之间两两恢复通行。

游戏中，小P和小Z拥有很多的修路工人，所以如果一个修路方案包含多条道路，那么这些道路可以同时开工。整个工程完工的时间就是这个工程中需要时间最长的道路的用时。

小P和小Z为了给你加大难度，一共要问你 q 个这样的问题。不同的问题之间不会互相影响，你可以认为这 q 个问题是发生在不同的平行世界中的。

2.2 输入格式

输入文件为road.in。

第一行包含三个整数 n, m, q 。

接下来 m 行，每行三个整数 a_i, b_i, c_i ，描述一条道路。注意道路的两端有可能是相同的城市。

接下来 q 行，每行描述一个问题：第一个数是这个问题的 k ，接下来 k 个数表示这次问题中需要两两恢复通行的城市编号。保证 k 至少为1；一个问题中可能多次出现同一个城市。

2.3 输出格式

输出文件为road.out。

输出 q 行，依次表示每一个问题的答案。

如果不需要建设任何道路，输出0；如果无论如何也无法完成，输出INF。

2.4 样例输入

```
5 6 3
1 2 4
2 3 4
3 1 4
1 4 3
2 4 3
3 4 3
3 1 2 3
4 1 2 3 5
2 5 5
```

2.5 样例输出

```
3
INF
0
```

2.6 样例解释

如果直接修复1到2和2到3的两条路，可使1,2,3三个城市两两恢复通行，需要4天。然而，如果修复1到4，2到4和3到4的三条路，只需3天就可以了。

由于5与其余城市都没有路，所以不可能使1,2,3,5四个城市恢复通行。然而，如果只需要5能到达自己，则不需要修复任何道路。

2.7 数据规模与约定

除样例外，所有的道路和问题都是完全随机生成的。

对于所有的数据， $1 \leq a_i, b_i \leq n, 0 \leq c_i \leq 10^9$ 。

下表中 $\sum k$ 表示该组数据当中所有问题的 k 值的总和。

测试点编号	n	m	q	$\sum k$
1	2	5	1	2
2	4	10	10	40
3	6	15	2	6
4	40	100	100	4000
5	80	200	3	80
6	120	300	300	36000
7	100	500000	4	100
8	200	500000	200000	1000000
9	300	500000	5	300
10	400	500000	200000	1000000
11	500	1000	6	500
12	1000	2000	200000	1000000
13	1500	3000	7	1500
14	2000	4000	200000	1000000
15	5000	10000	5000	20000
16	10000	50000	8	10000
17	20000	100000	9	20000
18	50000	200000	10	50000
19	100000	500000	200000	500000
20	200000	500000	200000	1000000

3 命名系统

3.1 题目描述

小P和小Z是一对非常好的朋友。今天认为现在人们的名字只有姓和名，信息量太少，所以他们决定研究一套新的人类命名系统。

这种命名系统的最基本原则是，一个人的名字中要包含尽可能多的关于他祖先的信息。小P和小Z喜欢父系，因此规定一个人的名字最好要以他父亲的名字结尾。（当然，也可以选择自己独特的命名方式）例如，小P和小Z的爷爷叫root，小P和小Z的父亲叫eggroot，小P叫peggroot，小Z叫zeggroot。我们把这种名字称为全名。

然而随着时间变化，人们的名字势必会越来越长，与辈分呈现线性的增长趋势，这是非常不好的。所以命名系统还给每个人从年老到年轻用正整数编号。例如爷爷、父亲、小P、小Z就分别被编上了1到4号。那么，一个人说自己的名字时，就可以只说自己的新名字，加上一个逗号，再加上父亲的编号。小P可以叫做p,2，小Z叫做z,2，他们的父亲可以叫做egg,1。我们把这种名字称为简化名。

灵活起见，简化名不一定要以父亲为结尾，也可以以更早的祖先为结尾，只要满足把逗号后的编号替换为该编号的全名，得到的名字是所要表达的全名即可。例如小P的简化名也可以是pegg,1，小Z也可以是zegg,1。

现在已经有 n 个人按照这个系统得到了名字，然而小P和小Z还想统计最常用的名字是什么。我们定义一个名字是常用的，是指这个名字在 n 个人的名字中，作为子串总计出现的次数至少为 m 。你的任务是确定这种名字是否存在。如果存在，你还要给出它的最大长度。

3.2 输入格式

输入文件为name.in。

第一行有两个正整数 n, m ，以空格隔开。

接下来 n 行，其中第 i 行表示编号为 i 的人的名字。这可能是全名或简化名。

保证所有人的全名都只包含小写字母且非空。

3.3 输出格式

输出文件为name.out。

输出最长的常用的名字的长度。如果没有任何常用的名字，输出0。

3.4 样例输入1

```
4 3
root
egg,1
p,2
zegg,1
```

3.5 样例输出1

```
7
```

3.6 样例输入2

```
2 5
aaa
a,1
```

3.7 样例输出2

```
2
```

3.8 样例解释

对于第一组样例，4个人的全名分别为：root，eggroot，peggroot，zeggroot。作为子串出现了至少3次的最长的常用名名字是eggroot，长度为7。

对于第二组样例，2个人的全名分别为aaa和aaaa。所求的常用名字是aa，出现了5次，长度为2。

3.9 数据规模与约定

设 L 表示输入文件中小写字母的总数，则：

对于前10%的数据， $n \leq 10, L \leq 20, m \leq 10$ 。

对于前25%的数据， $n \leq 50, L \leq 100, m \leq 1000$ 。

对于前40%的数据， $n \leq 200, L \leq 500, m \leq 100000$ 。

对于前55%的数据， $n \leq 500, L \leq 1000, m \leq 10^9$ 。

对于前70%的数据， $n \leq 50000, L \leq 100000, m \leq 10^{12}$ 。

对于前85%的数据， $n \leq 100000, L \leq 200000, m \leq 10^{15}$ 。

对于100%的数据， $1 \leq n \leq 200000, L \leq 500000, 1 \leq m \leq 10^{18}$ 。

对于后90%的数据，每3个数据中会出现1个不包含除了a以外的小写字母的数据（如样例2），1个 $n = 1$ 的数据和1个没有其他特征的数据。

所有数据是用某种方法随机生成的。