

嘎咕模拟赛

(请选手务必仔细阅读本页内容)

一、题目概况

中文题目名称	监狱	图	逃跑路径	矩形覆盖
英文题目名称	jail	graph	route	rect
可执行文件名	jail	graph	route	rect
输入文件名	jail.in	graph.in	route.in	rect.in
输出文件名	jail.out	graph.out	route.out	rect.out
提交文件名	jail.cpp	graph.cpp	route.cpp	rect.cpp
每个测试点时限	1秒	1秒	10秒	4秒
测试点数目	20	20	10	20
每个测试点分值	5	5	10	5
内存限制	1024MB	512MB	2048MB	1024MB
题目类型	传统题	传统题	传统题	传统题

二、编译命令

题目名称	jail	graph	route	rect
对于C++语言	-o jail jail.cpp -lm -std=c++14 -O2 -Wl,--stack=2147483647	-o graph graph.cpp -lm -std=c++14 -O2 -Wl,--stack=2147483647	-o route route.cpp -lm -std=c++14 -O2 -Wl,--stack=2147483647	-o rect rect.cpp -lm -std=c++14 -O2 -Wl,--stack=2147483647

三、注意事项

1. 文件夹名、文件名(程序名和输入输出文件名)必须使用英文小写。
2. C/C++中函数main()的返回值类型必须是int,程序正常结束时的返回值必须是0。
3. 统一评测时采用的机器配置为：windows下lemon评测。
4. 请尽力优化，会收获更多的部分得分。
5. **AK了不要大声喧哗，没AK也不要。**

监狱(jail)

题目描述

在 JOI 王国，安保最严格的地方就是 IOI 监狱。IOI 监狱中有 N 个房间，以 $1, \dots, N$ 编号。其中有 $N - 1$ 条通道。第 i ($1 \leq i \leq N - 1$) 条通道双向地连接房间 A_i 和 B_i 。任意两个房间都可以相互到达。

IOI 监狱中有 M 个囚犯，以 $1, \dots, M$ 编号。第 j ($1 \leq j \leq M$) 个囚犯的卧室和工作室分别是房间 S_j, T_j 。一个囚犯可能在另一个囚犯的卧室工作。然而，每个房间最多成为一个囚犯的卧室，一个囚犯的工作室。

一天早上，这 M 个囚犯需要从他们的卧室移动到他们的工作室。典狱长 APIO 先生需要按如下方式指示囚犯移动：

- **指令：**选择一个囚犯，然后命令他从当前所在的房间移动到一个与该房间有直接连边的房间。为了避免囚犯交流，不允许将囚犯移动到有其他囚犯在的房间。

为了尽早开始工作，APIO 先生想知道，是否存在一种给出任意条指令的方案使得每个囚犯以**最短路径**从卧室到达工作室。

请编写一个程序，在给定如上房间、通道和罪犯的所有信息后判断是否存在满足条件的方案。

输入格式

每个测试数据包包含多组测试用例。

第一行一个整数 Q ，表示这个测试数据包包含 Q 组测试用例。

对于每组测试用例：

第一行一个整数 N ，表示房间个数。

接下来 $N - 1$ 行每行两个整数 A_i, B_i 表示通道连接房间的编号。

接下来一行一个整数 M 表示囚犯个数。

接下来 M 行，每行两个整数 S_i, T_i 表示囚犯的卧室和工作室。

输出格式

输出 Q 行，第 i 行表示对于第 i 组测试用例，如果存在一种满足题意的方案，输出 **Yes**，否则输出 **No**。

样例

样例1输入

```
1
8
1 2
2 3
3 4
4 5
5 6
6 7
7 8
2
3 4
4 8
```

样例1输出

```
Yes
```

样例1解释

1. 让囚犯 2 从 4 号房间移动到 5 号房间。
2. 让囚犯 1 从 3 号房间移动到 4 号房间。
3. 让囚犯 2 从 5 号房间移动到 6 号房间。
4. 让囚犯 2 从 6 号房间移动到 7 号房间。
5. 让囚犯 2 从 7 号房间移动到 8 号房间。

样例2输入

```
2
7
1 2
2 3
3 4
4 5
3 6
6 7
2
4 1
5 7
4
1 2
1 3
1 4
3
2 3
3 4
4 2
```

样例2输出

```
Yes
No
```

数据范围

对于所有数据，满足：

- $1 \leq Q \leq 1000$ 。
- $1 \leq N \leq 120000$ 。
- $1 \leq A_i < B_i \leq N (i \in [1, N - 1])$ 。
- $2 \leq M \leq N$ 。
- $1 \leq S_i, T_i \leq N (i \in [1, M])$ 。
- $S_i (i \in [1, M])$ 互不相同。
- $T_i (i \in [1, M])$ 互不相同。
- $S_j \neq T_j (j \in [1, M])$ 。
- 任意两个房间之间可以通过给定道路互相到达。
- 对于所有测试用例， N 的总和不超过 120000。

测试点编号	特殊限制
1 ~ 2	$A_i = i, B_i = i + 1 (i \in [1, N - 1])$
3 ~ 6	$Q \leq 20, N \leq 250, M = 2$
7 ~ 10	$Q \leq 20, N \leq 250, M \leq 6$
11 ~ 12	$Q \leq 20, N \leq 250, M \leq 100$
13 ~ 14	$Q \leq 20, M \leq 500$
15 ~ 16	任意两个房间之间都可以通过不超过 20 条道路到达。
17 ~ 20	无

图(graph)

题目描述

给定一张 n 个点 m 条边的无向图，令 $k = \lceil \frac{m}{n-1} \rceil$ ，你需要判断能否找到两个不同的点 u, v ，满足它们之间存在 k 条**边不相交**路径，如果可以找到这样的 u, v ，你需要输出这些路径，如果存在多种构造方案，输出任意一种即可。

额外需要注意的是输入**可能存在重边**，也就是对于同一个无序对 (u, v) ，它们之间可能存在多条边，如果它们之间存在 s 条边那么你可以理解为这条边可以经过 s 次。

不过我们保证输入**不存在自环**。

输入格式

本题包含多组输入数据。

输入第一行一个正整数 T 表示数据组数。

对于每组输入数据，第一行输入两个正整数 n, m 表示点数和边数，接下来 m 行每行两个正整数 u, v 描述 u, v 间存在的一条边。

保证 $\sum n \leq 10^5$ ， $\sum m \leq 2 \times 10^5$ 。其中 $\sum n, \sum m$ 分别表示同一个测试点内所有输入数据的 n, m 之和。

输出格式

对于每组输入数据，如果不存在这样的 u, v ，那么输出一行一个整数 `-1`，否则先输出一行两个正整数 u, v 表示你找到的两个点，接下来输出 $k = \lceil \frac{m}{n-1} \rceil$ 行，每行第一个正整数 t 描述你选出来的路径长度，接下来 t 个正整数 x_1, x_2, \dots, x_t ，表示你选择了 $x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_t$ 这条路径，你需要保证 $x_1 = u$ 且 $x_t = v$ 。且你需要保证输出的 k 条路径满足边不相交的条件。

样例

样例1输入

```
3
3 1
1 3
4 7
1 2
2 3
3 4
4 1
1 3
2 4
1 4
5 5
1 2
2 3
3 4
4 5
3 5
```

样例1输出

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

数据范围

对于全部数据，满足 $1 \leq T \leq 10^4$, $2 \leq n \leq 10^5$, $1 \leq m \leq 2 \times 10^5$, $1 \leq u, v \leq n, u \neq v$ 。

测试点编号	约束
1 ~ 6	$m \leq 20, T \leq 5$
7 ~ 9	$n \leq 20, m \leq 100, T \leq 5$
10 ~ 14	$1 \leq n, m \leq 10^3$

测试点编号	约束
15 ~ 17	$k \leq 20$
18 ~ 20	无特殊限制

逃跑路径(route)

题目描述

IOI 王国使用 Byou（秒）作为时间单位，将一天划分成 S Byou，分别称为时刻 $0, 1, \dots, S - 1$ 。

IOI 王国中有 N 个城市和 M 条双向道路，均从 0 开始标号。保证任两个城市之间均连通。第 i 条道路连接城市 A_i 和 B_i ，需要恰好 L_i Byou 通过。每天的时刻 C_i 后，第 i 条道路将开始进行检查，直到当天结束。

JOI 组织是一个活跃在 IOI 王国中的秘密团体。出于其保密性，成员不能在道路上受到检查。如果其成员想要通过道路 i ，最晚要在时刻 $C_i - L_i$ 到达这条路的一端。道路的检查不会影响两端的城市。

现在有 Q 名 JOI 组织的成员，从 0 开始标号。第 j 名成员在某天的时刻 T_j ，要从城市 U_j 出发去城市 V_j 。成员可以在任意城市内停留任意长的时间。注意这名成员可能会在路上花费一天以上。

请计算每名成员花费的最短时间，精确到 Byou。

输入格式

第一行四个正整数 N, M, S, Q 。

接下来 M 行，第 i 行四个正整数表示 $A_{i-1}, B_{i-1}, L_{i-1}, C_{i-1}$ 。

接下来 Q 行，第 i 行三个正整数表示 $U_{i-1}, V_{i-1}, T_{i-1}$ 。

输出格式

输出共 Q 行，第 i 行表示第 $i - 1$ 名成员旅行所需的最短时间。

样例

样例1输入

```
4 5 20 6
0 1 3 19
0 2 2 8
1 2 4 15
1 3 5 14
2 3 1 18
0 3 5
0 3 7
0 3 9
2 0 6
3 1 10
1 2 15
```

样例1输出

```
3
8
14
2
5
7
```

样例2输入

```
6 10 100 9
5 3 4 29
1 0 6 26
0 4 2 7
0 5 18 18
2 0 79 82
3 4 35 46
1 2 15 57
2 4 3 6
4 1 21 83
3 2 47 53
0 2 63
0 4 70
0 4 98
0 5 25
0 5 19
0 4 96
0 5 2
0 3 62
0 3 83
```

样例2输出

```
42
32
4
93
99
6
102
60
39
```

样例3输入

```
8 12 1000000000000000 13
2 0 4451698272827 120985696255786
6 5 78520421713825 342652131468508
2 1 185377268405175 382583457603811
0 4 54350742205838 133614919589507
7 0 68486247989149 651590905094148
0 6 85177550834829 299184420663240
```

```
5 2 442329739732459 926608308293721
3 7 78020232822359 913548478810253
1 3 267796317244889 687571310475622
5 4 90590208828121 910324397566584
5 7 8414633059584 17796117322043
4 6 45682367792138 204548471584556
7 2 44779065000162
3 5 79376234836942
4 7 305556687070759
4 3 927935834343174
5 1 663284649258985
2 5 967584209777344
5 2 963749709374595
7 4 484562389171308
1 5 446160773830045
6 4 801452311055604
3 1 744524289545354
0 6 467418420721777
5 6 371181379240653
```

样例3输出

```
72937946261976
929038398222642
702857945988825
272921388674172
580895059624855
181808439529442
117602869946965
569788353034530
1181546234307589
244230056736534
513790925121797
617759130113052
674500988551485
```

数据范围

对于 100% 的数据，保证：

- $2 \leq N \leq 90$ 。
- $N - 1 \leq M \leq \frac{N(N - 1)}{2}$ 。
- $2 \leq S \leq 10^{15}$ 。
- $1 \leq Q \leq 3 \times 10^6$ 。
- $0 \leq A_i, B_i \leq N - 1$ 。
- $A_i \neq B_i$ 。
- $\forall i, j \in [0, M - 1]$, 若 $i \neq j$, 则有 $(A_i, B_i) \neq (A_j, B_j), (A_i, B_i) \neq (B_j, A_j)$ 。
- $1 \leq L_i \leq C_i < S$ 。
- 从任意城市出发走过一些边后，可以到达任意其他城市。
- $0 \leq U_j, V_j \leq N - 1$ 。
- $U_j \neq V_j$ 。
- $0 \leq T_j < S$ 。

每个测试点的具体限制见下表：

测试点编号	特殊性质
1	$N \leq 40, Q \leq 1000$
2 ~ 3	$N \leq 40, U_j = 0$
4	$N \leq 40$
5 ~ 7	$N \leq 60$
8 ~ 10	无

你可能需要较快的 OI 方式

矩形覆盖(rect)

题目描述

给定 $N \times M$ 的黑白矩阵。用尽可能少数量的矩形覆盖住所有黑色格子，要求：

- 每个黑色格子**恰好被一个**矩形覆盖；
- 任意两个矩形不重叠；
- 矩形不覆盖白色格子。

并输出方案。

输入格式。

第一行，两个正整数 N, M 。

接下来一个 $N \times M$ 的矩阵，每个位置是 C 或者 B。其中，C 代表黑色（克罗地亚语「crno」），B 代表白色（克罗地亚语「bijelo」）。

输出格式

输出 N 行，每行 M 个数，表示你的方案：

- 未被覆盖的区域，用 0 表示；
- 否则，设使用了 K 个矩形，将矩形用 $1 \sim K$ 标号后，对应位置用覆盖它的矩形编号表示。

每一行相邻的数要用空格隔开。

样例

样例1输入

```
4 5
CCBCB
CCBBB
CCCB
CCCB
```

样例1输出

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

样例2输入

```
7 5
CCCB
BCBB
BCCB
BCCB
CCCC
BBBB
BCCB
```

样例2输出

```
1 1 1 0 0
0 2 0 0 0
0 3 3 3 0
0 3 3 3 0
4 4 4 4 4
0 0 0 0 0
0 5 5 5 0
```

样例3输入

```
5 11
BBCCBCCBC
BCCBCBBCCC
CCCBCCCCC
BCBCCBCCB
CCCCBCBCCB
```

样例3输出

```
0 0 1 1 1 0 2 2 2 0 3
0 4 4 0 5 0 0 6 6 6 3
7 7 7 7 0 8 8 6 6 6 3
0 9 0 10 10 10 0 6 6 6 0
11 11 11 11 0 12 0 0 13 13 0
```

数据范围

对于 100% 的数据，保证 $1 \leq N, M \leq 500$ 。

测试点编号	$N, M \leq$
-------	-------------

测试点编号	$N, M \leq$
1 ~ 5	26
6 ~ 10	100
11 ~ 15	250
16 ~ 20	500

如果你输出的是最优解，得满分。

否则，设最优解用的矩形数量为 A ，你的解用的矩形数量为 B ，该测试点得分为 $0.75 \cdot (A/B)^{10} \cdot 5$ 分。