

NOIP 2016 提高组模拟赛 day2

题目名称	赛车比赛	物流运输	矩阵
可执行文件名	race	trans	matrix
输入文件名	race.in	trans.in	matrix.in
输出文件名	race.out	trans.out	matrix.out
每个测试点时限	2s	1s	3s
内存限制	256MB	256MB	400MB
测试点数目	20	10	20
每个测试点分值	5	10	5
是否有部分分	否	否	否
题目类型	传统型	传统型	传统型
是否有附加文件	否	否	否

提交源程序须加后缀

对于Pascal语言	race.pas	trans.pas	matrix.pas
对于C语言	race.c	trans.c	matrix.c
对于C++语言	race.cpp	trans.cpp	matrix.cpp

赛车比赛(race)

【题意描述】

USB 自己做了一辆卡丁车去参加 fl 赛事，经过了一轮预选赛，还剩下 n 名选手进入决赛。

由于各选手的预赛成绩不同，所以各选手的出发点 s_i 也是根据成绩而定的，有些人的出发点不同，有些人出发点相同。每位选手根据状态还有一个保持不变的速度 v_i 。为了简化问题，设跑道为一条数轴，选手的坐标即为其通过距离。

排名方法如下，如果一辆车在另一辆车前面，则这辆车在另一辆车前。如果两车的通过距离相同，则编号小的在前。

USB 的卡丁车是世界一流的，他不用担心当不了第一名。他现在想知道，第 t 时刻排在第 k 位的是那辆车。

【输入格式(race.in)】

第一行，包含一个正整数 n 。

第 $2 \sim n+1$ 行，第 $i+1$ 行包括两个正整数 v_i, s_i 。

第 $n+2$ 行，包含一个正整数 m 。

第 $n+3 \sim m+2$ 行，每行表示一个询问，包括两个正整数 t, k 。

【输出格式(race.out)】

输出包括 m 行，每行表示每个询问时刻 t 排在第 k 位的选手编号。

【样例输入】

```
4
2 100
3 50
4 60
5 1
4
1 1
50 2
60 4
100 1
```

【样例输出】

```
1
4
1
4
```

【数据规模与约定】

对于 30%的数据: $n, m \leq 1000$;

另有 40%的数据: $k=1$;

对于 100%的数据: $n, m \leq 7000; t \leq 1,000,000,000; v, s \leq 100,000; k \leq n$.

物流运输(trans)

物流公司要把一批货物从码头 A 运到码头 B。由于货物量比较大,需要 n 天才能运完。货物运输过程中一般要转停好几个码头。物流公司通常会设计一条固定的运输路线,以便对整个运输过程实施严格的管理和跟踪。由于各种因素的存在,有的时候某个码头会无法装卸货物。这时候就必须修改运输路线,让货物能够按时到达目的地。但是修改路线是一件十分麻烦的事情,会带来额外的成本。因此物流公司希望能够订一个 n 天的运输计划,使得总成本尽可能地小。

【输入格式(trans.in)】

文件 trans.in 的第一行是四个整数 n ($1 \leq n \leq 100$)、 m ($1 \leq m \leq 20$)、 K 和 e 。 n 表示货物运输所需天数, m 表示码头总数, K 表示每次修改运输路线所需成本。接下来 e 行每行是一条航线描述,包括了三个整数,依次表示航线连接的两个码头编号以及航线长度 (>0)。其中码头 A 编号为 1, 码头 B 编号为 m 。单位长度的运输费用为 1。航线是双向的。

再接下来一行是一个整数 d , 后面的 d 行每行是三个整数 P ($1 < P < m$)、 a 、 b ($1 \leq a \leq b \leq n$)。表示编号为 P 的码头从第 a 天到第 b 天无法装卸货物(含头尾)。同一个码头有可能在多个时间段内不可用。但任何时间都存在至少一条从码头 A 到码头 B 的运输路线。

保证数据有梯度。

【输出格式(trans.out)】

文件 trans.out 包括了一个整数表示最小的总成本。总成本= n 天运输路线长度之和+ K *改变运输路线的次数。

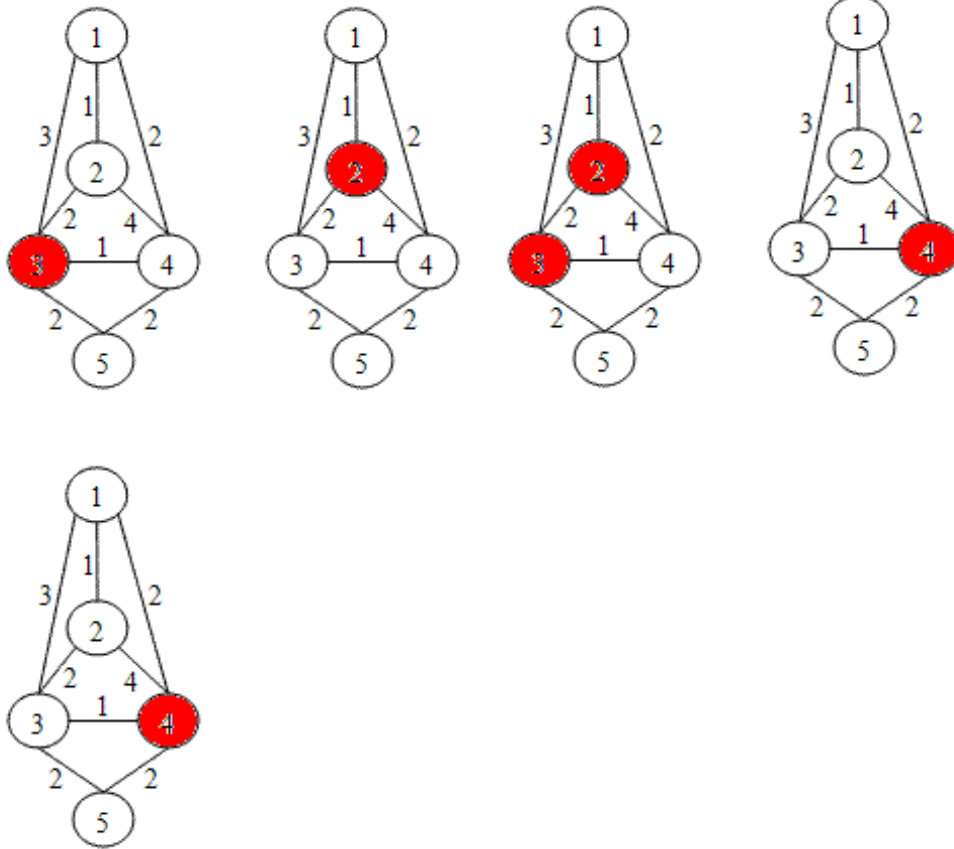
【样例输入】

```
5 5 10 8
1 2 1
1 3 3
1 4 2
2 3 2
2 4 4
3 4 1
3 5 2
4 5 2
4
2 2 3
3 1 1
3 3 3
4 4 5
```

【样例输出】

```
32
```

前三天走 1-4-5，后两天走 1-3-5，这样总成本为 $(2+2)*3+(3+2)*2+10=32$
 下图依次表示第一至五天的情况。红色表示不可用的码头



矩阵 (matrix)

【题意描述】

给定一个 $n \times n$ 的矩阵 (下标从 0 开始), 初始时所有格子的权值都是 0。

你需要支持 m 次操作, 每次操作可能是以下两种情况之一:

给一个子矩阵中的所有格子的权值加上一个给定的值。

询问一个子矩阵中所有格子的权值和, 答案对 q 取模。

【输入格式(matrix.in)】

第一行输入三个整数 n 和 m 和 q , 表示矩阵的边长和操作次数和模数。

接下来 m 行每行包含 5 个整数 $op, x1', x2', y1', y2'$ 。

本题采取强制在线措施:

令 $lastans$ 为之前所有 $op=2$ 的操作的答案“之和”【注意: 答案是对 q 取模的但是这个“之和”并没有对 q 取模, $lastans$ 初始为 0】

```
x1=(x1'+lastans)%n
```

```
x2=(x2'+lastans)%n
```

```
y1=(y1'+lastans)%n
```

```
y2=(y2'+lastans)%n
```

```
if(x2<x1)swap(x2,x1)
```

```
if(y2<y1)swap(y2,y1)
```

如果 $op=1$, 那么这个操作表示对以 $(x1,y1)$ 和 $(x2,y2)$ 为左下和右上端点的子矩阵中所有格子的权值加上 1。

如果 $op=2$, 那么这个操作表示询问以 $(x1,y1)$ 和 $(x2,y2)$ 为左下和右上端点的子矩阵中所有各自的权值和, 答案对 q 取模。

【输出格式(matrix.out)】

对于每一个 $op=2$ 的操作输出一行包含一个整数, 表示这个询问的答案, 答案对 q 取模。

【样例输入】

```
3 11 10
1 0 2 0 0
1 1 2 2 2
1 0 2 1 1
2 0 1 1 2
1 2 2 0 1
1 0 1 2 2
1 0 2 0 2
2 0 2 0 1
1 2 1 0 0
1 2 1 0 1
2 2 1 2 1
```

【样例输出】

3
4
0

【样例解释】

由于强制在线，实际上询问是这样的：

3 11 10
1 0 2 0 0
1 1 2 2 2
1 0 2 1 1
2 0 1 1 2
1 2 2 0 1
1 0 1 2 2
1 0 2 0 2
2 0 2 0 1
1 0 2 1 1
1 0 2 1 2
2 0 2 0 2

【数据规模与约定】

对于 80%的数据， $n \leq 200$ ， $m \leq 200$ 。

对于 95%的数据， $n \leq 3000$ ， $m \leq 10000$ 。

对于 100%的数据， $n \leq 8000$ ， $m \leq 100000$ ， $1 \leq q \leq 256$ 。

