

# NOIP 模拟题

## 一. 题目概况

中文题目名称	操作	天梯	距离
英文题目名	operate	coding	dist
可执行文件名	operate	coding	dist
输入文件名	operate.in	coding.in	dist.in
输出文件名	operate.out	coding.out	dist.out
每个测试点时限	1s	1s	2s
测试点数目	10	20	10
每个测试点分值	10	5	10
附加样例文件	无	有	无
结果比较方式	全文比较 (过滤行末空格及文末回车)		
题目类型	传统	传统	传统
运行内存上限	128M	256M	128M

## 二. 提交源程序文件名

对于 C++语言	operate.cpp	coding.cpp	dist.cpp
对于 C 语言	operate.c	coding.c	dist.c

## 三. 编译命令 (不包含任何优化开关)

对于 C++语言	<code>g++ -o operate operate.cpp -lm</code>	<code>g++ -o coding coding.cpp -lm</code>	<code>g++ -o dist dist.cpp -lm</code>
对于 C 语言	<code>g++ -o operate operate.c -lm</code>	<code>g++ -o coding coding.c -lm</code>	<code>g++ -o dist dist.c -lm</code>

### 注意事项:

- 1、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。

# 操作

(operate.cpp/c)

## 【问题描述】

Alex 给李华出了一道难题：

给出一个长度为  $n$  数组  $A_1, A_2, \dots, A_n$ ，初始值都为 0。

现在给出  $m$  个操作，请你输出操作后的序列。

操作共有两种：

1  $l$   $r$ ：将下标在区间  $[l, r]$  的元素的值都加 1。

这时，李华便嚷嚷道：“这不是水题吗？”

Alex：“别急别急，还有一个操作呢”

2  $l$   $r$ ：将下标在区间  $[l, r]$  的操作再执行一次。

## 【输入格式】

第一行两个正整数  $n, m$ ，表示数组长度与操作数量。

接下来  $m$  行，每行三个整数  $op\ l\ r$ ，表示一个操作，格式如题目描述所示。

操作编号从 1 开始。

对于操作 1，保证  $1 \leq l \leq r \leq n$ ；对于操作 2，保证  $1 \leq l \leq r < id$ ， $id$  为当前操作的编号。

## 【输出格式】

**由于数字可能很大，请将  $A_i$  对  $10^9 + 7$  取模。**

最后请你输出以下值：

$$ans = A_1 \text{ xor } A_2 \text{ xor } A_3 \text{ xor } \dots \text{ xor } A_n$$

输出仅一行，一个整数，表示  $ans$ 。

## 【样例输入输出】

[Input1]:

1 2

1 1 1

1 1 1

[Output1]:

2

[Input2]:

5 5

1 1 2

1 4 5

2 1 2

2 1 3

2 3 4

[Output2]:

7 7 0 7 7

**【数据范围】**

对于 20%的数据,  $n, m \leq 10$

对于 60%的数据,  $n, m \leq 2000$

对于 80%的数据,  $1 \leq n, m \leq 10^5$

对于 100%的数据,  $1 \leq n, m \leq 10^6$

# 天梯

(coding.cpp/c)

## 【题目描述】

解决了 Alex 的难题, 李华突然发现某 OJ 上的最新比赛: 天梯训练赛! 经过仔细的研究, 李华发现这场比赛的题目呈树形分布。AC 了第  $i$  道题后, 你可以解锁从属于它的  $m_i$  道题目  $x_{i,1}, x_{i,2} \dots x_{i,m_i}$ 。最开始只有第 1 道题是已解锁的。

李华决定解出这上面所有的题目, 根据他的评估, 解决第  $i$  道题需要花费他  $T_i$  小时的时间。然而李华刷题时间有限, 每一天只有  $h$  小时, 同时他不喜欢把题目留到第二天做。也就是说, 一道题必须完整地在一天内被解决。当然, 一天之内李华可以做好几道题。(如果时间允许)

同时, 刚学习了 DFS 的李华决定使用类似 DFS 的方式来做这些题, 具体来讲:

1. 有一个栈  $S$ , 初始时只包含第 1 道题。
2. 弹出栈顶的题目  $x$ 。
3. 花费  $T_x$  小时完成这一道题; 如果今天时间不够了, 就到明天再完成。
4. 完成后, 会解锁  $m_x$  道题目, 将它们以**某一顺序**压入栈中, 回到第 2 步骤。

李华想要你帮他算一算他完成这些题目所需要的最少天数。

## 【输入格式】

第一行包括两个正整数  $n$  和  $h$ , 表示题目数量与李华一天能够刷题的时间。

第二行包括  $n$  个由空格分隔的正整数  $T_1, T_2, T_3 \dots T_n$ , 表示李华解决每一道题分别需要的时间。

接下来  $n$  行描述题目的结构。第  $x$  行首先包括一个非负整数  $m_x$ , 然后是  $m_x$  个正整数, 表示解决第  $x$  题后会解锁的  $m_x$  个题目的编号  $x_{i,1}, x_{i,2} \dots x_{i,m_i}$ 。

## 【输出格式】

输出仅一行, 表示李华解决所有题目所需要的最少天数。

## 【样例输入输出】

[Input1]:

5 24

13 24 22 12 16

1 3

0

2 2 5

0  
1 4

[Output1]:

5

[Input2]:

10 8  
1 4 3 1 7 3 2 2 4 4  
3 2 5 10  
2 3 4  
0  
0  
1 6  
3 7 8 9  
0  
0  
0  
0

[Output2]:

4

[Input3]:

见附加文件 coding.in。

[Output3]:

见附加文件 coding.ans。

#### 【样例解释】

在第一组数据中，每一天最多都只能完成一道题，因此做完所有题需要 5 天。

对于第二组数据，李华可以按照以下顺序完成题目：

第一天：完成第 1、5 题

第二天：完成第 6、9 题

第三天：完成第 7、8、10 题

第四天：完成第 2、3、4 题

【数据范围】

测试点编号	$n$	$m_i$	约定
1	$= 1$	$= 0$	$1 \leq T_i \leq h \leq 24$
2	$\leq 10$	$\leq 5$	
3			
4			
5	$\leq 100$	$\leq 1$	
6		$\leq 3$	
7	$\leq 300$	$\leq 5$	
8			
9	$\leq 500$	$\leq 8$	
10	$\leq 1000$		
11	$\leq 2000$		
12			
13	$\leq 3000$	$\leq 9$	
14			
15	$\leq 4000$	$\leq 10$	
16			
17			
18			
19	$\leq 5000$		
20			

# 距离

(dist.cpp/c)

## 【题目描述】

李华最近在研究图论问题。

李华有一个普通的图，由 $n$ 个点与 $m$ 条有长度的单向边组成。

Alex 挑选了一些边，把边上的数字改成了 $x$ ， $x$ 可以指代任意的正整数。Alex 指定了 $x$ 的不同取值，问李华此时 $S$ 点到 $T$ 点的最短路径长度是多少。

李华忙着去做今天的作业，他想请你帮忙求出当 $x$ 的取值不同时， $S$ 点到 $T$ 点的最短路径长度有多少种以及这些最短路径长度之和。

## 【输入格式】

第一行包括两个正整数 $n$ 和 $m$ ，表示图的点数与边数。

第2行至第 $m+1$ 行每行包括 $u, v, w$ ，表示有一条边从 $u$ 指向 $v$ ，长度为 $w$ 。若该边为长度可变的边， $w$ 用“ $x$ ”表示。

第 $m+2$ 行一个正整数 $q$ ，表示询问数。

接下来 $q$ 行，每行两个整数 $S, T$ ，表示起点与终点。

## 【输出格式】

输出 $q$ 行，每行两个整数，表示最短路径数量与长度之和。若最短路径有无数种，输出“inf”。

## 【样例输入输出】

[Input1]:

4 4

1 2 x

2 3 x

3 4 x

1 4 8

3

2 1

1 3

1 4

[Output1]:

0 0  
inf  
3 17

[Input2]:

3 5  
3 2  $x$   
2 1  $x$   
2 1 5  
1 3 10  
3 1 20  
6  
1 2  
2 3  
3 1  
2 1  
3 2  
1 3

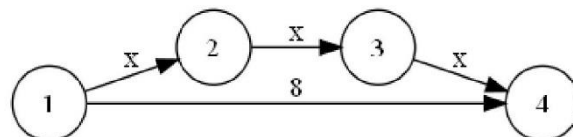
[Output2]:

inf  
5 65  
15 185  
5 15  
inf  
1 10

**【样例解释】**

对于第一组数据:

1. 不存在从 2 到 1 的路径
2. 对于任意的正整数  $x$ , 从 1 到 3 的最短路都是  $2x$
3. 从 1 到 4 的最短路可以是  $3(x = 1)$ ,  $6(x = 2)$ , 或者  $8(x \geq 3)$





**【数据范围】**

对于 50%的数据,  $1 \leq n \leq 30, 1 \leq m \leq 300, w \leq 50$

对于 100%的数据,  $1 \leq n \leq 500, 0 \leq m \leq 10000, 1 \leq w \leq 10^6, 1 \leq q \leq 10$