

## CCF 全国信息学奥林匹克联赛（NOIP2014）复赛

## 提高组 day2

（请选手务必仔细阅读本页内容）

## 一. 题目概况

中文题目名称	无线网路发射器选址	寻找道路	解方程
英文题目与子目录名	wireless	road	equation
可执行文件名	wireless	road	equation
输入文件名	wireless.in	road.in	equation.in
输出文件名	wireless.out	road.out	equation.out
每个测试点时限	1 秒	1 秒	1 秒
测试点数目	10	10	20
每个测试点分值	10	10	5
附加样例文件	有	有	有
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	128M	128M

## 二. 提交源程序文件名

对于 C++语言	wireless.cpp	road.cpp	equation.cpp
对于 C 语言	wireless.c	road.c	equation.c
对于 pascal 语言	wireless.pas	road.pas	equation.pas

## 三. 编译命令（不包含任何优化开关）

对于 C++语言	g++ -o wireless wireless.cpp -lm	g++ -o road road.cpp -lm	g++ -o equation equation.cpp -lm
对于 C 语言	gcc -o wireless wireless.c -lm	gcc -o road road.c -lm	gcc -o equation equation.c -lm
对于 pascal 语言	fpc wireless.pas	fpc road.pas	fpc equation.pas

## 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。
- 3、全国统一评测时采用的机器配置为：CPU AMD Athlon(tm) 64x2 Dual Core CPU 5200+，2.71GHz，内存 2G，上述时限以此配置为准。
- 4、只提供 Linux 格式附加样例文件。
- 5、特别提醒：评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以其为准。

## 1. 无线网络发射器选址

(wireless.cpp/c/pas)

### 【问题描述】

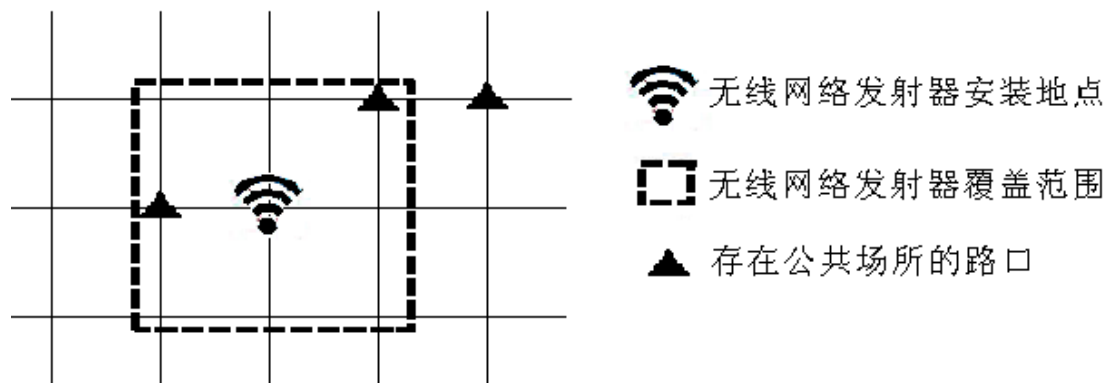
随着智能手机的日益普及，人们对无线网的需求日益增大。某城市决定对城市内的公共场所覆盖无线网。

假设该城市的布局为由严格平行的 129 条东西向街道和 129 条南北向街道所形成的网格状，并且相邻的平行街道之间的距离都是恒定值 1。东西向街道从北到南依次编号为 0,1,2...128,南北向街道从西到东依次编号为 0,1,2...128。

东西向街道和南北向街道相交形成路口，规定编号为  $x$  的南北向街道和编号为  $y$  的东西向街道形成的路口的坐标是  $(x,y)$ 。在某些路口存在一定数量的公共场所。

由于政府财政问题，只能安装一个大型无线网络发射器。该无线网络发射器的传播范围是一个以该点为中心，边长为  $2*d$  的正方形。传播范围包括正方形边界。

例如下图是一个  $d=1$  的无线网络发射器的覆盖范围示意图。



现在政府有关部门准备安装一个传播参数为  $d$  的无线网络发射器，希望你帮助他们在城市内找出合适的安装地点，使得覆盖的公共场所最多。

### 【输入】

输入文件名为 wireless.in。

第一行包含一个整数  $d$ ，表示无线网络发射器的传播距离。

第二行包含一个整数  $n$ ，表示有公共场所的路口数目。

接下来  $n$  行，每行给出三个整数  $x, y, k$ ，中间用一个空格隔开，分别代表路口的坐标  $(x, y)$  以及该路口公共场所的数量。同一坐标只会给出一次。

### 【输出】

输出文件名为 wireless.out。

输出一行，包含两个整数，用一个空格隔开，分别表示能覆盖最多公共场所的安装地点方案数，以及能覆盖的最多公共场所的数量。

## 【输入输出样例】

wireless.in	wireless.out
1	1 30
2	
4 4 10	
6 6 20	

## 【数据说明】

对于 100% 的数据,  $1 \leq d \leq 20$ ,  $1 \leq n \leq 20$ ,  $0 \leq x \leq 128$ ,  $0 \leq y \leq 128$ ,  $0 < k \leq 1,000,000$ 。

## 2. 寻找道路

(road.cpp/c/pas)

## 【问题描述】

在有向图  $G$  中, 每条边的长度均为 1, 现给定起点和终点, 请你在图中找一条从起点到终点的路径, 该路径满足以下条件:

1. 路径上的所有点的出边所指向的点都直接或间接与终点连通。
2. 在满足条件 1 的情况下使路径最短。

注意: 图  $G$  中可能存在重边和自环, 题目保证终点没有出边。

请你输出符合条件的路径的长度。

## 【输入】

输入文件名为 road.in。

第一行有两个用一个空格隔开的整数  $n$  和  $m$ , 表示图有  $n$  个点和  $m$  条边。

接下来的  $m$  行每行 2 个整数  $x$ 、 $y$ , 之间用一个空格隔开, 表示有一条边从点  $x$  指向点  $y$ 。

最后一行有两个用一个空格隔开的整数  $s$ 、 $t$ , 表示起点为  $s$ , 终点为  $t$ 。

## 【输出】

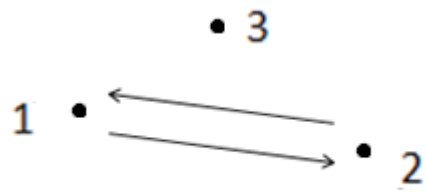
输出文件名为 road.out。

输出只有一行, 包含一个整数, 表示满足题目描述的最短路径的长度。如果这样的路径不存在, 输出 -1。

## 【输入输出样例 1】

road.in	road.out
3 2	-1
1 2	
2 1	
1 3	

【输入输出样例说明】

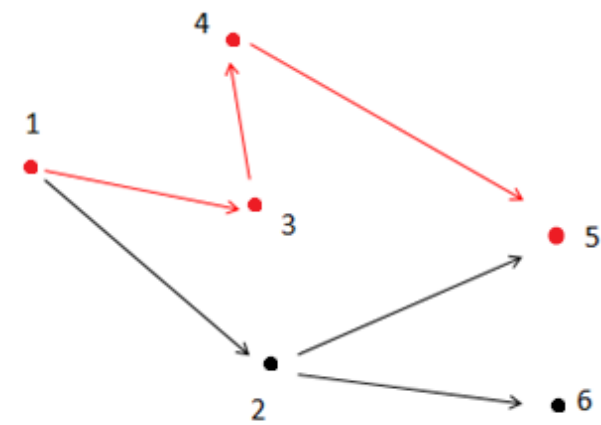


如上图所示，箭头表示有向道路，圆点表示城市。起点 1 与终点 3 不连通，所以满足题目描述的路径不存在，故输出-1。

【输入输出样例 2】

road.in	road.out
6 6 1 2 1 3 2 6 2 5 4 5 3 4 1 5	3

【输入输出样例说明】



如上图所示，满足条件的路径为 1->3->4->5。注意点 2 不能在答案路径中，因为点 2 连了一条边到点 6，而点 6 不与终点 5 连通。

【数据说明】

- 对于 30%的数据， $0 < n \leq 10$ ， $0 < m \leq 20$ ；
- 对于 60%的数据， $0 < n \leq 100$ ， $0 < m \leq 2000$ ；
- 对于 100%的数据， $0 < n \leq 10,000$ ， $0 < m \leq 200,000$ ， $0 < x, y, s, t \leq n$ ， $x \neq t$ 。

3. 解方程

(equation.cpp/c/pas)

【问题描述】

已知多项式方程：

$$a_0 + a_1x + a_2x^2 + \cdots + a_nx^n = 0$$

求这个方程在 $[1, m]$ 内的整数解（ $n$  和  $m$  均为正整数）。

【输入】

输入文件名为 equation.in。

输入共  $n+2$  行。

第一行包含 2 个整数  $n$ 、 $m$ ，每两个整数之间用一个空格隔开。

接下来的  $n+1$  行每行包含一个整数，依次为 $a_0, a_1, a_2, \dots, a_n$ 。

【输出】

输出文件名为 equation.out。

第一行输出方程在 $[1, m]$ 内的整数解的个数。

接下来每行一个整数，按照从小到大的顺序依次输出方程在 $[1, m]$ 内的一个整数解。

【输入输出样例 1】

equation.in	equation.out
2 10	1
1	1
-2	
1	

【输入输出样例 2】

equation.in	equation.out
2 10	2
2	1
-3	2
1	

【输入输出样例 3】

equation.in	equation.out
2 10	0
1	
3	
2	

**【数据说明】**

对于 30% 的数据， $0 < n \leq 2$ ， $|a_i| \leq 100$ ， $a_n \neq 0$ ， $m \leq 100$ ；

对于 50% 的数据， $0 < n \leq 100$ ， $|a_i| \leq 10^{100}$ ， $a_n \neq 0$ ， $m \leq 100$ ；

对于 70% 的数据， $0 < n \leq 100$ ， $|a_i| \leq 10^{10000}$ ， $a_n \neq 0$ ， $m \leq 10000$ ；

对于 100% 的数据， $0 < n \leq 100$ ， $|a_i| \leq 10^{10000}$ ， $a_n \neq 0$ ， $m \leq 1000000$ 。

## 花团(bag)

时间限制: 3.000 Sec 内存限制: 256 MB

### 题目描述

物品集合  $S$  初始为空, 按时间递增顺序依次给出  $q$  次操作, 操作如下:

- \* 1  $v\ w\ e$  表示在  $S$  中加入一个体积为  $v$  价值为  $w$  的物品, 第  $e$  次操作结束之后移除该物品。
- \* 2  $v$  表示询问。你需要回答:
  1. 当前  $S$  是否存在一个子集使得子集中物品体积和为  $v$ 。
  2. 当前  $S$  的所有物品体积和为  $v$  的子集中, 价值和最大是多少 (空集的价值和为 0)。

### 输入

第一行三个正整数  $q, \max v, T$  表示操作数、最大可能的  $v$ 、及是否强制在线。

接下来  $q$  行每行描述一个操作。

设修正值  $d = T \times \text{lastans}$ 。这里  $\text{lastans}$  表示上次询问的\*\*两个答案的异或和\*\*, 若没有上次则  $\text{lastans} = 0$ 。

则第  $i$  行中, 第一个整数  $op$  表示操作类型;

若操作类型为 1, 则接下来三个整数  $v', w', e'$  表示加入一个体积为  $v_i = v' - d$ , 价值为  $w_i = w' - d$ , 在第  $e_i = e' - d$  时间后被移除的物品;

若操作类型为 2, 则接下来一个整数  $v'$  表示询问  $v_i = v' - d$ 。

一行中相邻整数以单个空格分隔。

### 输出

对于每个询问 (2 类型的操作) 输出一行两个整数  $e$  和  $\max w$ , 由空格分隔。

若不存在这样的子集,  $e = \max w = 0$ ;

否则  $e = 1$ ,  $\max w$  为最大的价值和。

### 样例输入

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

19 20 1
1 2 19 11
2 2
1 27 27 22
1 20 34 36
2 29
1 9 8 9
1 5 19 8
1 1 15 19
2 3
1 36 40 54
1 37 50 52
2 40
2 62
1 1 17 16
1 1 7 16
1 13 16 18
1 9 11 19
2 10
2 34

1 19
0 0
1 34
1 46
0 0
1 26
0 0
```

## 样例输出



1 0  
1 8  
1 9  
1 7  
0 0  
1 3  
0 0  
  
1 19  
0 0  
1 34  
1 46  
0 0  
1 26  
0 0

提示

对于所有数据， $1 \leq q \leq 15000, 1 \leq v_i \leq \max v \leq 15000, 0 \leq w_i \leq 15000, i \leq e_i \leq q$ 。

对于每个测试点，若问题 1 回答全对，则得 40% 的分数；若问题 2 回答全对，则另得 60% 的分数。每个子任务的得分是其中各测试点的最小值。  
注意，即使你只回答了一个问题，每次仍要输出两个整数，以免答案错位。

Subtask #	分值	$q, v$ 的限制	$T$ 的限制
1	15	$q \leq 18, v \leq 15000$	$T = 0$
2	20	$q, v \leq 1000$	$T \in \{0, 1\}$
3	20	$q, v \leq 6000$	$T = 0$
4	20	$q, v \leq 10000$	$T = 0$
5	25	$q, v \leq 15000$	$T \in \{0, 1\}$