

复赛模拟一

(请选手务必仔细阅读本页内容)

一. 题目概况

中文题目名称	放养	狂奔	奶牛线路	检查列表
英文题目与子目录名	stocking	Stampede	Routing	Checklist
可执行文件名	stocking	Stampede	Routing	Checklist
输入文件名	stocking.in	Stampede.in	Routing.in	Checklist.in
输出文件名	stocking.out	Stampede.out	Routing.out	Checklist.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
测试点数目	10	11	12	10
每个测试点分值	10	9	8	10
附加样例文件	有	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	传统

二. 提交源程序文件名

对于 C++ 语言	stocking.cpp	Stampede.cpp	Routing.cpp	Checklist.cpp
-----------	--------------	--------------	-------------	---------------

三. 运行内存限制

内存上限	128M	128M	128M	128M
------	------	------	------	------

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、特别提醒: 评测在 NOI Linux 下进行

1 放养

(stocking)

一个鲜为人知的事实是，奶牛拥有自己的文字：「牛文」。
牛文由 26 个字母 a 到 z 组成，但是当奶牛说牛文时，可能与我们所熟悉的 abcdefghijklmnopqrstuvwxyz 不同，她会按某种特定的顺序排列字母。
为了打发时间，奶牛 Bessie 在反复哼唱牛文字母歌，而 Farmer John 好奇她唱了多少遍。
给定一个小写字母组成的字符串，为 Farmer John 听到 Bessie 唱的字母，计算 Bessie 至少唱了几遍完整的牛文字母歌，使得 Farmer John 能够听到给定的字符串。

Farmer John 并不始终注意 Bessie 所唱的内容，所以他可能会漏听 Bessie 唱过的一些字母。
给定的字符串仅包含他记得他所听到的字母。

输入

输入文件名为 `stocking.in`。

输入的第一行包含 26 个小写字母，为 a 到 z 的牛文字母表顺序。

下一行包含一个小写字母组成的字符串，为 Farmer John 听到 Bessie 唱的字母。

输出

输出文件名为 `stocking.out`。

输出 Bessie 所唱的完整的牛文字母歌的最小次数。

字符串的长度不小于 1 且不大于 1000。

Bear and Big Brother.in	Bear and Big Brother.out
abcdefghijklmnopqrstuvwxy z mood	3

2 狂奔

(Stampede .cpp)

农夫约翰的 N 头奶牛正在约翰农场前方的道路上狂奔，实际上它们是在赛跑，看看哪头牛跑的最快。

从上方看，每头奶牛可由一个单位长度的水平线段表示，每条线段用其左端点在 $t=0$ 时的坐标位置指定。

例如， $(-3,6)$ 将指定在 $t=0$ 时，由 $(-3,6)$ 到 $(-2,6)$ 的线段表示的奶牛。

每头奶牛以一定的速率向右 ($+x$ 方向) 移动，该速率由其向右移动 1 个单位长度所花费的整数时间指定。

约翰对于这些产奶不积极，就知道撒欢乱跑的奶牛十分不满。

他打算在它们的比赛结束后，严厉的训诫他们。

为了确定哪些奶牛参加了比赛，约翰将自己定位在 $(0,0)$ 处，并沿着一条向 $+y$ 方向延伸的射线观察。

随着比赛的进行，约翰会看到沿着这条射线可见的第一头奶牛，

也就是说，如果一头奶牛在穿过约翰视线的整个过程中，如果始终都有牛挡在它的前面被约翰看到，那么它就不会被约翰看见。

请计算约翰在正常比赛中，可看见的奶牛数量。

输入格式

输入文件名为 Stampede.in。

第一行包含整数 N 。

接下来 N 行，每行包含三个整数 x,y,r ，表示一头奶牛在 $t=0$ 时刻，其对应线段的左端点

位于 (x,y) ，它沿 $+x$ 方向的移动速度是每 r 单位时间移动 1 单位距离。

输出格式

输出从 $t=0$ 开始，整个比赛期间约翰可以看到的奶牛数量。

Stampede.in	Stampede.out
3 -2 1 3 -3 2 3 -5 100 1	2

样例解释

FJ 可以看到牛 1 和 2 但看不到牛 3

【数据范围】

$1 \leq N \leq 50000$,

$-1000 \leq x \leq -1$,

$1 \leq y \leq 10^6$, 数据保证不同奶牛的 y 互不相同。

$1 \leq r \leq 10^6$

3 奶牛线路

(Routing.cpp)

厌倦了农场寒冷的冬季天气，奶牛贝茜计划飞往一个温暖的目的地度假。

不幸的是，她发现只有一家航空公司，博维尼亚航空，愿意向奶牛出售机票，而且这些机票的结构有些复杂。

博维尼亚航空公司拥有 N 架飞机，每架飞机都在由两个或多个城市组成的特定“航线”上飞行。

例如，一架飞机可能从城市 1 起飞，然后飞到城市 5，然后飞到城市 2，最后飞到城市 8。

没有一个城市会在一条航线上出现多次。

如果贝茜选择了一条航线，那么她可以在航线上的任何城市登机，然后在航线上的任何城市下飞机。

她不需要在航线的第一个城市登机或在最后一个城市下机。

每条航线都有一定的费用，只要贝茜乘坐了某个航线，不论乘坐时途径的城市有多少，都需要支付全部的航线费用。

如果贝茜在旅途中多次搭乘某个航线（也就是说，如果她离开了该航线，后来又从另一个城市回来乘坐该航线），那么每次乘坐该航线时，她都必须为此付费。

贝茜想找到从她所在的农场（在 A 市）到热带目的地（B 市）的最便宜的旅行方式。

请帮助她确定她需要支付的最低费用是多少，以及为达到该最低费用所需的最少搭乘航班次数。

输入格式

第一行包含三个整数 A, B, N 。

接下来 $2N$ 行，每两行描述一条航线，第一行包含航线的乘坐费用以及航线途径的城市数量。第二行包含按航线顺序排列的城市列表。

输出格式

在一行输出贝茜从 A 到 B 所需要支付的最低费用以及所需的最少搭乘航班次数。

如果无法到达目的地，则输出 $-1 -1$ 。

Routing.in	Routing.out
3 4 3	22
3 5	
1 2 3 4 5	
2 3	
3 5 4	
1 2	
1 5	

数据范围

$1 \leq N \leq 1000$,

$1 \leq$ 航线费用 $\leq 10^9$,

$1 \leq$ 航线途径城市数量 ≤ 100 ,

城市编号范围 $[1,1000]$ 。

4 检查列表

(Checklist.cpp)

每天，农夫约翰都会走过牧场，检查每头奶牛的健康状况。
 他的农场中有两种奶牛，荷斯坦奶牛和根西岛奶牛。
 其中，荷斯坦牛 H 头，编号 $1 \sim H$ 。根西岛牛 G 头，编号 $1 \sim G$ 。
 每头奶牛都位于二维平面中的某个点上（不一定互不相同）。
 约翰希望检查路线从荷斯坦牛 1 开始，到荷斯坦牛 H 结束，并且沿途能够检查完所有奶牛。
 为了方便起见，他将对检查过的奶牛进行记录。
 他希望按照编号的顺序来检查荷斯坦牛以及根西岛牛。
 也就是说，所有奶牛检查完毕后，他得到的长度为 $H+G$ 的奶牛检查顺序序列，应该满足所有荷斯坦牛构成的子序列（不一定连续）应为 $1 \cdots H$ ，所有根西岛牛构成的子序列（不一定连续）应为 $1 \cdots G$ 。
 也就是说所有 $H+G$ 头奶牛的顺序序列，应该是将编号顺序为 $1 \cdots H$ 的荷斯坦牛序列与编号顺序为 $1 \cdots G$ 的根西岛牛序列交错排列而成。
 当约翰从一头奶牛处移动到另一头相距为 D 的奶牛处时，他会消耗 D^2 能量。
 请你计算，约翰按上述说明，访问完所有奶牛所需消耗的能量最低是多少。

输入格式

第一行包含两个整数 H 和 G 。

接下来 H 行，每行包含两个整数 x 和 y ，表示一头荷斯坦牛所处位置的横纵坐标。

再接下来 G 行，每行包含两个整数 x 和 y ，表示一头根西岛牛所处位置的横纵坐标。

输出格式

输出一个整数，表示约翰访问完所有奶牛所需消耗的最低能量。

数据范围：

$1 \leq H, G \leq 1000$,

$0 \leq x, y \leq 1000$

。

Checklist.in	Checklist.out
3 2	20
0 0	
1 0	
2 0	
0 3	
1 3	