CCF 全国信息学奥林匹克联赛（NOIP2018）复赛

提高组 模拟赛强化训练

（请选手务必仔细阅读本页内容）

一．题目概况

中文题目名称	城市规划	Count	水壶
英文题目与子目录名（Lemon 测评机请忽略）	city	count	kettle
可执行文件名	city	count	kettle
输入文件名	city.in	count.in	kettle.in
输出文件名	city.out	count.out	kettle.out
每个测试点时限	1 秒	1 秒	5 秒
测试点数目	10	10	15
附加样例文件	见附件	见附件	见附件
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	256M	256M	512M

二． 交源程序文件名

对于 C++语言	city.cpp	count.cpp	kettle.cpp
对于 C 语言	city.c	count.c	kettle.c
对于 pascal 语言	city.pas	count.pas	kettle.pas

三．编译命令（不包含任何优化开关）

对于 C++语言	g++ -o citycity.cpp -lm	g++ -o countcount.cpp -lm	g++ -o kettlekettle.cpp -lm
对于 C 语言	gcc -o citycity.c -lm	gcc -o countcount.c -lm	gcc -o kettlekettle.c -lm
对于 pascal 语言	fpc city.pas	fpc count.pas	fpc kettle.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、测评环境为 Window10

4、特别提醒：评测在 Lemon 下进行，各语言的编译器版本以其为准。

1．城市规划

(city.cpp/c/pas)

【问题描述】

最近比特镇正在迅速建成。沿着美丽的大街，一座座新建筑拔地而起。小 Q 喜欢沿着大街走，但问题是不同的建筑位于街对面。为了从一个建筑到另一个建筑，有时需要通过漫长的步行穿过最近的人行道。所以他决定写一个程序，计算如何沿着大街平移所有人行道，使得人行道的布局最有利于行人。他希望尽可能多的人行道出现在某些建筑物的前面，同时人行道的移动距离应当是最小的。

大街以直线表示，人行道被视为这条线上的点。所有建筑物都平行于大街，所以你可以认为它们是直线上的一条条线段。每条线段都具有左边界和右边界。如果某人行道位于某建筑物的左右边界之间（包括边界点），则你可以认为该人行道位于该建筑物的前方。由于人行道已经按照某些标准建立，小 Q 决定保持它们之间的距离，所以他想将所有的人行道移动相同的距离。

请帮助小 Q 写一个程序计算最优布局

【输入】

第一行包含两个正整数 n,m(1<=n<=10000,1<=m<=1000)，分别表示人行道和建筑的个数。

第二行包含 n 个整数 ai(0<=ai<=10^6)，分别表示每条人行道的坐标，可能存在两条人行道重合。

接下来 m 行，每行两个整数 li,ri(0<=li<ri<=10^6)，分别表示每座建筑的左右边界，这些线段可以相互重叠。

【输出】

输出一行两个整数 d 和 s，其中 d 表示平移距离的绝对值，s 表示出现在至少一座建筑物前面的人行道个数。你需要输出 s 最大的解，若有多个 d 使得 s 最大，那么输出 d 最小的解。注意你可以向左或者向右平移人行道。

【输入输出样例 1】

city.in	**city.out**
4 2 1 6 6 1 4 5 3 5	1 2

【数据范围】

对于 30%的数据 1<=n<=1000,1<=m<=100

对于 100%的数据，1<=n<=10000,1<=m<=1000

2. Count

(count.cpp/c/pas)

【问题描述】

给一个长为 n 的序列 $A_1, A_2, \ldots, A_n$，定义 (i, j)（规定 $i < j$）为好点对，当且仅当满足下列条件之一：

- $i = j - 1$
- $\forall k \in (i, j), A_k < min(A_i, A_j)$

现在有 m 组询问，每组询问给定一个区间，求这个区间内的好点对的个数。

给定一个 $Type$，当 $Type = 0$ 的时候不强制在线，否则强制在线。具体操作请看输入格式。

【输入】

输入第一行 3 个正整数 n , m , $Type$，分别表示序列长度，询问的个数，输入数据的种类。

输入第二行 n 个正整数，第 i 个数表示 A_i。

接下来 m 行，每行两个非负整数 l，r。当 $Type = 0$ 的时候，询问区间就是 $[l, r]$，否则令 $u = (l + last - 1) \mod n + 1$，$v = (r + last - 1) \mod n + 1$，那么当前询问区间就是 $[min(u, v), max(u, v)]$。其中 $last$ 是上一次询问的答案，初始时 $last = 0$。

【输出】

输出共 m 行，每行非负整数，第 i 个数表示第 i 次询问的答案。

【输入输出样例】

count.in	count.out
3 2 0 2 1 2 1 1 1 3	0 3

【数据范围】

对于 30%的数据，M,N<=3*10^2,Ai<=10^9

对于 100%的数据，M,N<=3*10^5,Ai<=10^9

3. 水壶

(kettle.cpp/c/pas)

【问题描述】

JOI 君所居住的 IOI 市以一年四季都十分炎热著称。

IOI 市被分成 H 行，每行包含 W 块区域。每个区域都是建筑物、原野、墙壁之一。
IOI 市有 P 个区域是建筑物，坐标分别为 (A1,B1), (A2,B2),,…, (AP,BP)。
JOI 君只能进入建筑物与原野，而且每次只能走到相邻的区域中，且不能移动到市外。

JOI 君因为各种各样的事情，必须在各个建筑物之间往返。虽然建筑物中的冷气设备非常好，但原野上太阳非常毒辣，因此在原野上每走过一个区域都需要 1 升水。此外，原野上没有诸如自动售货机、饮水处之类的东西，因此 IOI 市的市民一般都携带水壶出行。大小为 x 的水壶最多可以装 x 升水，建筑物里有自来水可以将水壶装满。
由于携带大水壶是一件很困难的事情，因此 JOI 君决定携带尽量小的水壶移动。因此，为了随时能在建筑物之间移动，请你帮他写一个程序来计算最少需要多大的水壶。

现在给出 IOI 市的地图和 Q 个询问，第 i 个询问包含两个整数 Si, Ti，对于每个询问，请输出：要从建筑物 Si 移动到 Ti，至少需要多大的水壶？

【输入】

第一行四个空格分隔的整数 H,W,P,Q。
接下来 H 行，第 i 行有一个长度为 W 的字符串，每个字符都是 . 或 # 之一，. 表示这个位置是建筑物或原野，# 表示这个位置是墙壁。
接下来 P 行描述 IOI 市每个建筑物的位置，第 i 行有两个空格分隔的整数 Ai 和 Bi，表示第 i 个建筑物的位置在第 Ai 行第 Bi 列。保证这个位置在地图中是 .。
接下来 Q 行，第 i 行有两个空格分隔的整数 Si, Ti。

【输出】

输出 Q 行，第 i 行一个整数，表示要从建筑物 Si 移动到 Ti，至少需要多大的水壶。如果无法到达，输出 -1。如果不需要经过原野就能到达，输出 0。

【输入输出样例 1】

kettle.in	kettle.out
5 5 4 4 ....##. .#... ..#.. ..... 1 1 4 2 3 3 2 5 1 2 2 4 1 3	3 4 4 2

3 4	

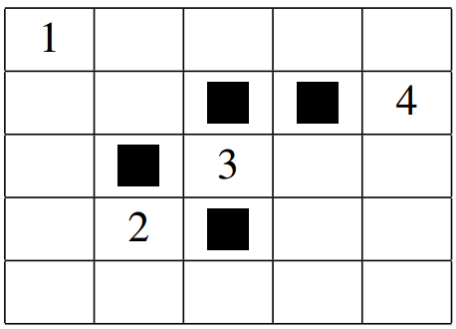
【数据范围】

对于所有数据，$1 \leq H, W \leq 2000, 2 \leq P \leq 2 \times 10^5, 1 \leq Q \leq 2 \times 10^5, 1 \leq A_i \leq H, 1 \leq B_i \leq W, (A_i, B_i) \neq (A_j, B_j)$ $(1 \leq i < j \leq P), 1 \leq S_i < T_i \leq P (1 \leq i \leq Q)$。

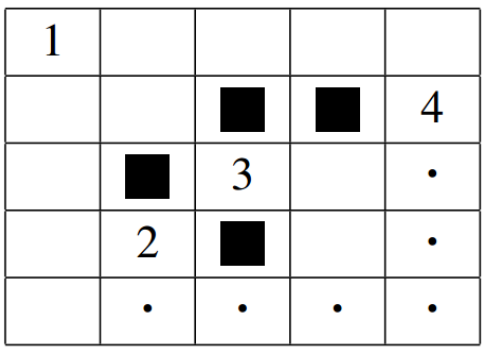
子任务编号	分值	附加条件
1	10	$H, W, P \leq 200$
2	30	$P \leq 5000, Q = 1$
3	30	$P \leq 5000, Q \leq 10^4$
4	30	

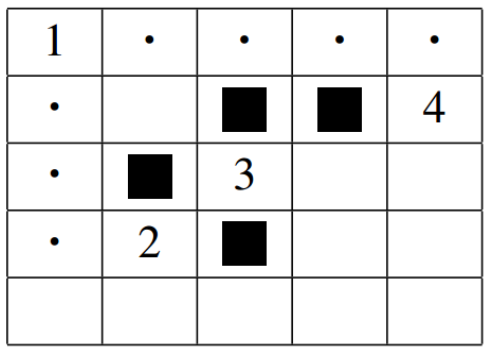
【样例解释】

初始状态如下：



其中■表示墙，含有数字的区域表示建筑，其他区域为原野。





如果只不经过建筑物（左图），则需要容量为 6 升的水壶；但如果经过建筑物 1（右图），从建筑物 2 到 1 需要 3 升水，而从建筑物 1 到 4 需要 4 升水，因此只需要容量为 4 升的水壶。