

全国青少年信息学奥林匹克竞赛

CCF-NOIP-2018

提高组（复赛）模拟试题

中文题目名称	贪吃蛇	分糖果	排序
英文题目名称	snake	candy	sort
输入文件名	snake.in	candy.in	sort.in
输出文件名	snake.out	candy.out	sort.out
每个测试点时限	1 秒	1 秒	3 秒
内存限制	512MB	512MB	512MB
测试点数目	20	20	20
每个测试点分值	5	5	5
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

提交源程序须加后缀

对于 Pascal 语言	snake.pas	candy.pas	sort.pas
对于 C 语言	snake.c	candy.c	sort.c
对于 C++ 语言	snake.cpp	candy.cpp	sort.cpp

注意：最终测试时，所有编译命令均不打开任何优化开关。

贪吃蛇

【问题描述】

贪吃蛇是一个好玩的游戏。在本题中，你需要对这个游戏进行模拟。

这个游戏在一个 n 行 m 列的二维棋盘上进行。我们用 (x, y) 来表示第 x 行第 y 列的格子，那么左上角为 $(1, 1)$ ，右下角为 (n, m) 。

我们用一个长度为 k 的不重复的坐标的序列（形如 $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$ ）来表示一条长度为 k 的蛇，其中 (x_1, y_1) 称为蛇的头部。在游戏的任何时刻，都满足 $k > 1$ 。

游戏开始时，蛇的长度为 1，坐标为 (x_s, y_s) 。接下来会进行 q 个操作，每个操作是以下两种类型之一：

- 1 d : 蛇的头部往 d 方向伸长一格，其中 d 为 UDLR 之一，分别表示“上”、“下”、“左”、“右”。
- 2: 蛇的尾部缩短一格，保证该操作前蛇的长度大于 1。

棋盘上还有 t ($0 \leq t < nm$) 个障碍物，分别位于 (u_i, v_i) ($1 \leq i \leq t$)，保证没有两个障碍物占据同一个格子

在任何时候，如果蛇的头部碰到蛇的身体（即蛇的其他格子），或碰到棋盘上的障碍物，或移动到棋盘的边界之外，那么蛇会立即死亡。

你的任务是，输入游戏配置以及 q 个操作，判断蛇是否会死亡。

【输入格式】

输入的第一行包含四个非负整数 n, m, t, q ，具体含义见问题描述。

接下来 t 行，每行两个整数，表示一个障碍物的坐标。保证每个坐标都在棋盘上，即在 $(1, 1)$ 到 (n, m) 之间，且不存在重复的坐标。

接下来一行两个整数，表示游戏开始时蛇的坐标 (x_s, y_s) 。保证该坐标在棋盘上，且不是任何一个障碍物的坐标。

接下来 q 行，每行给出一个操作，具体格式和含义见问题描述。

【输出格式】

如果在 q 个操作后蛇没有死亡，输出 -1，否则输出一个整数 k ，表示蛇在第 k 个操作之后死亡。

【样例输入 1】

```
3 4 2 10
1 3
3 3
1 1
1 D
1 R
2
1 R
2
1 R
1 U
1 L
1 L
1 L
```

【样例输出 1】

```
8
```

【样例说明 1】

第 8 个操作时，蛇的头部碰到了位于(1, 3)的障碍物，因此在这个操作之后死亡。

【样例输入 2】

```
2 5 0 6
1 2
1 R
1 R
1 D
1 L
1 U
1 L
```

【样例输出 2】

```
5
```

【样例说明 2】

第 5 个操作时，蛇的头部碰到了位于(1, 3)的蛇的身体，因此在这个操作之后死亡。

【数据规模与约定】

在所有测试点中，有 20% 的测试点 $n = 1$ 。

在所有测试点中，有 40% 的测试点 $t = 0$ 。

在所有测试点中，有 20% 的测试点满足任何时候蛇的长度不超过 2。

以上三类特殊的测试点可能存在交叉。

对于全部测试点， $1 \leq n, m \leq 100$ ， $0 \leq t < nm$ ， $1 \leq q \leq 10000$ 。

分糖果

【问题描述】

到了学期末，在幼儿园工作的刘老师要为自己所带班级的小朋友分发糖果。刘老师的班上共有 n 名小朋友，第 i 位小朋友对糖果的喜爱程度为 a_i ，他在本学期的表现评分为 b_i 。刘老师分配糖果的方法如下：

1. 以某个顺序安排这 n 位小朋友排成一排，刘老师从头到尾逐一分配糖果。
2. 队伍中的第 i 位小朋友至少获得的糖果数量为前 i 位小朋友对糖果的喜爱程度之和。
3. 由于第 i 位小朋友可以看见第 $i-1$ 位小朋友获得的糖果数量，为了不让第 i 位小朋友觉得不公平，刘老师保证第 i 位小朋友获得的糖果不少于第 $i-1$ 位小朋友。
4. 在为第 i 位小朋友分配完糖果后，刘老师将额外再奖励第 i 位小朋友数量为 b_i 的糖果。

我们设第 i 位小朋友获得的糖果数量为 c_i ，形式化地讲：

$$c_i = \begin{cases} a_1 + b_1 & i = 1 \\ \max\{c_{i-1}, \sum_{j=1}^i a_j\} + b_i & 2 \leq i \leq n \end{cases}$$

由于预算有限，刘老师希望你能帮她安排这 n 位小朋友的顺序，使得获得糖果最多的小朋友，所获得的糖果数量尽可能少。

【输入格式】

第一行包含一个正整数 T ，表示测试数据的组数。

接下来描述这 T 组测试数据，每组数组的第一行包含一个正整数 n ，表示刘老师班上小朋友的数量。

每组数据接下来 n 行中，每行两个正整数，分别为 a_i 和 b_i ，含义如问题描述中所述。

【输出格式】

共 T 行，每行包含一个整数，表示被分配到最多糖果的那位小朋友最少获得的糖果数量。

【样例输入 1】

```
1
3
4 1
2 2
1 2
```

【样例输出 1】

8

【样例说明 1】

按照 1、2、3 排列队伍，获得最多糖果的小朋友获得糖果数量为 10；
按照 1、3、2 排列队伍，获得最多糖果的小朋友获得糖果数量为 9；
按照 2、1、3 排列队伍，获得最多糖果的小朋友获得糖果数量为 9；
按照 2、3、1 排列队伍，获得最多糖果的小朋友获得糖果数量为 8；
按照 3、1、2 排列队伍，获得最多糖果的小朋友获得糖果数量为 9；
按照 3、2、1 排列队伍，获得最多糖果的小朋友获得糖果数量为 8。
当按照 3、2、1 这样排列队伍时，三位小朋友所对应的 a_i 与 b_i 如下：

$$(1, 2)、(2, 2)、(4, 1)$$

第 1 位小朋友获得的糖果为 $1 + 2 = 3$ ；

第 2 位小朋友获得的糖果为 $\max\{3, 1 + 2\} + 2 = 5$ ；

第 3 为小朋友获得的糖果为 $\max\{5, 1 + 2 + 4\} + 1 = 8$ 。

【样例输入 2】

1
12
9 68
18 45
52 61
39 83
63 67
45 99
52 54
82 100
23 54
99 94
63 100
52 68

【样例输出 2】

902

【数据规模与约定】

所有测试点的数据规模与约定如下：

测试点编号	n 的规模	T 的规模	约定
1	$n = 1$	$T = 1$	/
2	$n = 2$		
3	$n = 5$	$T = 5$	
4	$n = 9$		
5	$n = 15$		
6	$n = 15$		
7	$n = 16$		
8	$n = 16$		
9	$n = 3,000$	$T = 10$	$a_i = b_i$
10	$n = 3,000$		$b_i = a_i + 1$
11	$n = 5,000$		
12	$n = 5,000$		/
13	$n = 10,000$		
14	$n = 10,000$		
15	$n = 20,000$		
16	$n = 20,000$		
17	$n = 30,000$		
18	$n = 30,000$		
19	$n = 50,000$		
20	$n = 50,000$		

对于全部测试数据满足： $1 \leq a_i, b_i \leq 10^9$ 。

排序

【问题描述】

已知一个正整数数组中包含 n 个正整数，依次为 a_1, a_2, \dots, a_n 。

我们将进行 m 次操作。对于第 j 次操作，会指定一个位置 p_j ，将所有位置 k 满足 $p_j \leq k \leq n$ 且大小满足 $a_k \leq a_{p_j}$ 的正整数 a_k 从数组中拿出，并将这些正整数按照从小到大的顺序进行排序，之后重新放回数组中。

举一个例子，对于正整数数组 1 4 2 5 3 而言，若选择位置 $p_j = 2$ ，则拿出的正整数为 4 2 3，分别对应 $k = 2, k = 3, k = 5$ ，拿出的正整数排序以后变为 2 3 4，再将它们放回到数组中，数组变成 1 2 3 5 4。

在每次操作以后，你需要回答整个数组中逆序对的总数。正整数 a_i 与 a_j 构成一个逆序对，当且仅当 $a_i > a_j$ 且 $1 \leq i < j \leq n$ 。

【输入格式】

输入第一行包含两个正整数 n 和 m ，其中 n 表示数组的长度， m 表示操作的次数。

接下来一行包括 n 个正整数，依次表示正整数数组中的元素 a_1, a_2, \dots, a_n 。

接下来一行包括 m 个正整数，依次表示询问的位置 p_1, p_2, \dots, p_m 。

【输出格式】

输出文件共包括 m 行，其中第 j 行表示第 j 次操作以后整个数组的逆序对总数。

【样例输入 1】

```
5 3
1 4 2 5 3
5 2 4
```

【样例输出 1】

```
3
1
0
```

【样例输入 2】

```
7 4
7 7 1 4 2 5 3
6 4 2 1
```


【样例输出 2】

12
10
5
0

【数据规模与约定】

所有测试点的数据规模与约定如下：

测试点编号	n 的规模	m 的规模	a_i 的规模	特殊限制				
1	$n = 2$	$m = 1$	$1 \leq a_i \leq n$	$p_1 = n$				
2	$n = 5000$							
3	$n = 300,000$							
4	$n = 400,000$							
5	$n = 500,000$							
6	$n = 400$	$m = n$		$1 \leq a_i \leq n$	无			
7	$n = 500$							
8	$n = 10,000$	$m = 500$				$1 \leq a_i \leq n$	无	
9	$n = 20,000$							
10	$n = 30,000$							
11	$n = 250,000$	$m = n$	$1 \leq a_i \leq 10^9$					数据为随机生成
12	$n = 300,000$							
13	$n = 350,000$							
14	$n = 400,000$							
15	$n = 450,000$							
16	$n = 500,000$							
17	$n = 350,000$		$m = n$	$1 \leq a_i \leq 10^9$	无			
18	$n = 400,000$							
19	$n = 450,000$							
20	$n = 500,000$							

第 11 ~ 16 号测试点的生成方法如下：

奇数号测试点的每个 a_i 在区间 $[1, n]$ 中等概率随机生成，偶数号测试点的每个 a_i 在区间 $[1, 10^9]$ 中等概率随机生成。

这 6 个测试点的每个 p_j 都在区间 $[1, n]$ 中等概率随机生成。

对于全部测试数据，满足 $1 \leq n \leq 500,000$ ， $1 \leq m \leq 500,000$ ， $1 \leq a_i \leq 10^9$ ， $1 \leq p_j \leq n$ 。