

NOIP 模拟赛

时间：2024 年 7 月 26 日 08:00 ~ 12:00

题目名称	幸运数字	道路交通	二叉树	最短路
题目类型	传统型	传统型	传统型	传统型
目录	luck	traffic	tree	path
可执行文件名	luck	traffic	tree	path
输入文件名	luck.in	traffic.in	tree.in	path.in
输出文件名	luck.out	traffic.out	tree.out	path.out
每个测试点时限	1.5 秒	1.0 秒	2.0 秒	2.0 秒
内存限制	32 MiB	128 MiB	256 MiB	256 MiB
测试点数目	20	10	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	luck.cpp	traffic.cpp	tree.cpp	path.cpp
-----------	----------	-------------	----------	----------

编译选项

对于 C++ 语言	-O2 -std=c++14
-----------	----------------

注意事项

- 文件名 (包括程序名和输入输出文件名) 必须使用英文小写。
- C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 `0`。
- 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
- 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
- 程序可使用的栈空间大小与该题内存空间限制一致。
- 每道题目所提交的代码文件大小限制为 100KB。
- 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。
- 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。

幸运数字 (luck)

【题目描述】

小 U 记录下来了连续 n 天的彩票中奖数字 (每天一个), 第 i 天的中奖数字是 a_i , 他坚信其中存在某种规律。

具体来说, 小 U 确定了一个数 $l (l < n)$, 他打算研究从第 $x (x \leq n - l + 1)$ 天开始的连续 l 个中奖数字, 即 $a_x, a_{x+1}, \dots, a_{x+l-1}$ 。他定义第 $x (x \leq n - l + 1)$ 天和第 $y (y \leq n - l + 1)$ 天的差异为满足 $a_{x+i} \neq a_{y+i} (0 \leq i < l)$ 的 i 的数量。

小 U 现在有 q 个问题, 对于每个问题他给出了一个整数 k_i , 他想让你帮他求出来对于每一天来说, 有多少天和这一天的差异不超过 k_i (不包括这一天本身)。

注意, 你不需要也不应该考虑第 $n - l + 2$ 到第 n 天。

【输入格式】

从文件 **luck.in** 中读入数据。

第一行包含两个整数 n, l 。

第二行包含 n 个整数 a_1, a_2, \dots, a_n 。

第三行包含一个整数 q 。

接下来 q 行, 每行一个整数 k_i 。

【输出格式】

输出到文件 **luck.out** 中。

共 q 行。对于每个询问, 输出一行 $n - l + 1$ 个整数, 第 i 个整数表示有多少天与第 i 天的差异不超过 k_i 。

【样例 1 输入】

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

【样例 1 输出】

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

【样例 2】

见选手目录下的 *luck/luck2.in* 与 *luck/luck2.ans*。
该组样例满足测试点 1 ~ 5 的限制。

【样例 3】

见选手目录下的 *luck/luck3.in* 与 *luck/luck3.ans*。
该组样例满足测试点 6 ~ 9 的限制。

【样例 4】

见选手目录下的 *luck/luck4.in* 与 *luck/luck4.ans*。
该组样例满足测试点 17 ~ 20 的限制。

【测试点约束】

对于全部测试数据， $1 \leq n \leq 10^4$, $1 \leq a_i \leq 10^9$, $1 \leq q \leq 100$, $0 \leq k_i \leq l$ 。

测试点编号	$n \leq$	q	附加限制
1 ~ 5	300	$\leq 10^2$	无
6 ~ 9	2,000		
10 ~ 13	10^4	$= 1$	$k_1 = 0$
14 ~ 16			无
17 ~ 20		$\leq 10^2$	

【提示】

请注意本题特殊的内存限制。

道路交通 (traffic)

【题目描述】

如果将 C 国的平面图画在平面直角坐标系上,那么它可看作一个以 $(0,0),(A,0),(0,B),(A,B)$ 为四个顶点的矩形。

在 C 国中有 n 座城市,第 i 座城市的坐标为 (x_i,y_i) 。如果 $x_i = 0$,我们就称第 i 座城市在最西侧;如果 $x_i = A$,就称它在最东侧。

在这些城市间修建了 m 条笔直的道路,由于交通管理需要,每条道路可能是双行道(双向通行)也可能是单行道(单向通行),且道路不在城市以外的任何地方相交。换句话说,将城市看成点,道路看成连接两点的**线段**,则没有任何两条线段在非端点处相交。

现在请你求出,对于最西侧的每一座城市,能够沿着道路到达多少座最东侧的城市。

【输入格式】

从文件 **traffic.in** 中读入数据。

第一行包含四个整数 n,m,A,B 。

接下来 n 行,每行两个整数 x_i,y_i 。

接下来 m 行,每行三个整数 c_i,d_i,k_i ,表示一条连接第 c_i 座城市和第 d_i 座城市间的道路。如果 $k_i = 1$,那么这条道路是只能从 c_i 前往 d_i 的单行道;如果 $k_i = 2$,那么这条道路是双行道。

【输出格式】

输出到文件 **traffic.out** 中。

输出若干行,请按照 y_i **从大到小**的顺序输出每一个最西侧城市对应的答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 2
2 0
3 2
```

【样例 2 输入】

```
1 12 13 7 9
2 0 1
3 0 3
4 2 2
5 5 2
6 7 1
7 7 4
8 7 6
9 7 7
10 3 5
11 0 5
12 0 9
13 3 9
14 1 3 2
15 3 2 1
16 3 4 1
17 4 5 1
18 5 6 1
19 9 3 1
20 9 4 1
21 9 7 1
22 9 12 2
23 10 9 1
24 11 12 1
25 12 8 1
26 12 10 1
```

【样例 2 输出】

```
1 4
2 4
3 0
4 2
```

【测试点约束】

对于 30% 的测试数据，有 $n, m \leq 6000$ 。

对于 100% 的数据，有 $1 \leq n \leq 3 \times 10^5, 0 \leq m \leq 9 \times 10^5, 1 \leq A, B \leq 10^9, 0 \leq x_i \leq A, 0 \leq y_i \leq B, 1 \leq c_i, d_i \leq n, k_i \in \{1, 2\}$ ，保证最西侧的城市至少有一个，保证每个无序对 $\{c_i, d_i\}$ 只会出现一次。

二叉树 (tree)

【题目描述】

Asuka 面前有一棵带点权的二叉树， i 号点点权为 a_i ，1 号点是它的根。

Asuka 对于优雅的二叉搜索树情有独钟。另外，这棵树的点权时不时会被修改。

你需要完成两种操作。第一个操作为修改某个点的点权。第二个操作为回答某棵子树内有多少棵子树为二叉搜索树。

一棵二叉搜索树中每个点都满足左子树中所有点权都小于等于自己，且右子树中所有点权都大于等于自己。

【输入格式】

从文件 *tree.in* 中读入数据。

第一行三个整数 n, q, K ，其中 K 表示测试点编号。对于样例有 $K = 0$ 。

接下来 n 行每行两个整数表示每个点的左儿子和右儿子编号。编号为 0 表示没有。

接下来一行 n 个整数表示初始点权 a_i 。

接下来 q 行，每行形如 $1\ x\ y$ 或 $2\ x$ 。

- $1\ x\ y$ 表示修改 x 号点点权为 y 。
- $2\ x$ 表示查询 x 的子树中有多少颗子树是二叉搜索树。

【输出格式】

输出到文件 *tree.out* 中。

对于每个询问操作，输出一行一个整数表示答案。

【样例 1 输入】

```
1 6 5 0
2 2 3
3 4 0
4 5 6
5 0 0
6 0 0
7 0 0
8 4 1 3 2 2 5
9 2 2
10 1 3 3
11 1 2 2
```

```
12 1 3 5
13 2 1
```

【样例 1 输出】

```
1 1
2 5
```

【样例 2】

见选手目录下的 *tree/tree2.in* 与 *tree/tree2.ans*。
此组样例满足测试点 8 ~ 10 的限制。

【样例 3】

见选手目录下的 *tree/tree3.in* 与 *tree/tree3.ans*。
此组样例满足测试点 16 ~ 20 的限制。

【子任务】

对于全部的测试数据， $1 \leq n, q \leq 2 \times 10^5, 1 \leq a_i, x, y \leq n$ 。

测试点编号	$n, q \leq$	特殊性质
1, 2	10^2	A
3 ~ 5	5,000	
6, 7	2×10^5	A,B
8 ~ 10		A
11 ~ 15		B
16 ~ 20		无

特殊性质 A: 第 i 个点的父亲在 $[1, i - 1]$ 中随机，若儿子位置满则一直随机。
特殊性质 B: 询问满足 $x = 1$ 。

最短路 (path)

【题目描述】

$n \times m$ 个点排列成了一个 n 行 m 列的方阵，我们记 (x, y) 表示第 x 行的第 y 个点。同时，第 y ($1 \leq y \leq m$) 列还有一个参数 c_y 。

对于每个点 (x, y) :

- 如果 $x < n$ ，则 (x, y) 向 $(x + 1, y)$ 有一条有向边，边权为 c_y 。
- 如果 $y < m$ ，则 (x, y) 向 $(x, y + 1)$ 有一条有向边，边权为 x^2 。

有 q 个独立的询问，每次询问给出一个点 (x_i, y_i) ，请你求出从 $(1, 1)$ 到 (x_i, y_i) 的最短路。

【输入格式】

从文件 *path.in* 中读入数据。

输入的第一行包含两个整数 n, m 。

第二行包含 m 个整数 c_1, c_2, \dots, c_m 。

第三行包含一个整数 q 。

接下来 q 行每行包含两个整数 x_i, y_i ，表示一次询问。

【输出格式】

输出到文件 *path.out* 中。

共 q 行，对于每次询问，输出一行一个整数表示答案。

【样例 1】

见选手目录下的 *path/path1.in* 与 *path/path1.ans*。

【样例 2】

见选手目录下的 *path/path2.in* 与 *path/path2.ans*。

【测试点约束】

对于全部测试数据， $1 \leq x_i \leq n \leq 10^9, 1 \leq y_i \leq m \leq 2 \times 10^5, q \leq 2 \times 10^5, c_y \leq 10^9$ 。

测试点编号	$n \leq$	$m \leq$	特殊性质
1 ~ 3	2,000	2,000	无
4 ~ 8	10^9	2×10^5	A
9 ~ 15	2×10^5		无
16 ~ 20	10^9		

特殊性质 A: $c_2 > c_3 > \cdots > c_m$ 。