

NOIP 模拟赛

题目名称	题目类型	输入文件	输出文件	时间限制	空间限制
跑操	传统题	run.in	run.out	1.0 秒	512 Mib
排序	传统题	sort.in	sort.out	1.0 秒	512 Mib
洪水	传统题	flood.in	flood.out	1.0 秒	512 Mib
流量	传统题	flow.in	flow.out	2.0 秒	512 Mib

编译选项

```
1 | -O2 -std=c++14 -static
```

搬题人：DaiRuiChen007

跑操 (run)

【题目描述】

小 D 的学校的操场上有 $n + 1$ 个学生在跑操，他们的编号依次为 $0, 1, \dots, n$ 。

操场可以被看成一根无限长的一维数轴，对于第 i 个人，其初始位置在 $-i$ 上。

每秒内，学生会按 $0, 1, \dots, n$ 的顺序依次按如下规则移动：

- 第 0 个学生每秒向正方向移动 1 的距离。
- 第 i 个学生 ($1 \leq i \leq n$) 如果与第 $i - 1$ 的学生的距离 $> d_i$ ，那么他会移动到第 $i - 1$ 个学生的上一个位置。

小 D 突然想到了 q 个询问，第 i 个询问需要你求出第 t_i 秒在 $[l_i, r_i]$ 范围内的学生数量。

你能回答小 D 的 q 个询问吗？

【输入格式】

从 `run.in` 中读入数据。

第一行两个整数 n, q 。

第二行 n 个整数，第 i 个正整数表示 d_i 。

接下来 q 行，每行三个整数，分别表示 t_i, l_i, r_i 。

【输出格式】

输出到 `run.out` 中。

输出 q 行，每行一个整数，第 i 行表示第 i 次询问的答案。

【样例 1 输入】

1	3	2
2	1	2 2
3	1	1 5
4	2	0 1

【样例 1 输出】

1	1
2	2

【样例 1 解释】

四个学生的位置变化如下：

- 第 0 秒： $[0, -1, -2, -3]$ 。
- 第 1 秒： $[1, 0, -2, -3]$ 。
- 第 2 秒： $[2, 1, 0, -1]$ 。

【样例 2】

见下发文件中的 `run2.in` 与 `run2.ans`。

【样例 3】

见下发文件中的 `run3.in` 与 `run3.ans`。

【数据范围】

对于所有测试数据有： $1 \leq n, q \leq 5 \times 10^5, 1 \leq d_i, t_i \leq 10^9, 0 \leq l_i \leq r_i \leq 10^9$ 。

子任务编号	分值	特殊限制
Subtask 1	5	$n, q, t_i, l_i, r_i \leq 10$
Subtask 2	25	$n, q, t_i, l_i, r_i \leq 1000$
Subtask 3	70	无特殊限制

排序 (sort)

【题目描述】

小 D 正在学习排序。

他有一个长度为 n 的 01 序列 a_1, a_2, \dots, a_n 。

每一秒，他会找到所有 $a_i = 0$ 且 $a_{i+1} = 1$ 的所有 i ，并**同时**交换所有的 a_i, a_{i+1} 。

请你帮小 D 求出 m 秒后的序列。

【输入格式】

从 `sort.in` 中读入数据。

第一行两个整数 n 和 m 。

第二行 n 个整数表示 a_1, a_2, \dots, a_n 。

【输出格式】

输出到 `sort.out` 中。

一行 n 个整数表示 m 秒后的序列。

【样例 1 输入】

```
1 10 3
2 1 0 0 1 0 1 1 1 0 1
```

【样例 1 输出】

```
1 1 1 1 0 1 0 1 0 1 0
```

【样例 1 解释】

第 1 秒后，整个序列为：[1, 0, 1, 0, 1, 0, 1, 1, 1, 0]。

第 2 秒后，整个序列为：[1, 1, 0, 1, 0, 1, 0, 1, 1, 0]。

【样例 2】

见下发文件中的 `sort2.in` 与 `sort2.ans`。

【样例 3】

见下发文件中的 `sort3.in` 与 `sort3.ans`。

【数据范围】

对于所有测试数据有： $1 \leq n, m \leq 2 \times 10^6, a_i \in \{0, 1\}$ 。

子任务编号	分值	特殊限制
Subtask 1	5	$n, m \leq 3000$
Subtask 2	5	没有两个 1 相邻
Subtask 3	30	$n, m \leq 10^5$
Subtask 4	60	无特殊限制

洪水 (flood)

【题目描述】

小 D 的家乡有 n 个城市，所有城市通过 $n - 1$ 条道路相连，保证任意两个城市之间都可以通过道路互相到达。

现在发生了 m 场洪水，第 i 场洪水发生在 p_i ，水位为 h_i 。

为了防止洪水泛滥，市长决定给每条边上修一个水坝，第 i 条边上的水坝高度为 w_i 。

对于一条边 (u_i, v_i) ，若 u_i 上产生一场水位为 h 的洪水且 $h > w_i$ ，那么 v_i 上也会产生一场水位为 h 的洪水。

小 D 想知道是否存在一个城市没有被洪水淹没过。

但这个问题太简单了，小 D 一眼就做出了这道题，于是他把这个题加强了一下，现在他想知道：

如果第 i 条边的水坝高度从 $[l_i, r_i]$ 中等概率随机选取一个整数作为 w_i ，那么存在一个城市没有被洪水淹没的概率是多少？

【输入格式】

从 `flood.in` 中读入数据。

第一行两个整数 n 和 m 。

接下来 $n - 1$ 行，每行表示一条边，每行四个整数 u_i, v_i, l_i, r_i ，表示边的两个端点和权值的范围。

接下来 m 行，每行两个整数 p_i, h_i ，表示一场洪水。

【输出格式】

输出到 `flood.out` 中。

一行一个整数表示答案。

【样例 1 输入】

```
1 5 2
2 1 2 1 10
3 2 3 2 9
4 1 4 3 12
5 2 5 4 6
6 1 7
7 5 5
```

【样例 1 输出】

1	888437475
---	-----------

【样例 2】

见下发文件中的 flood2.in 与 flood2.ans。

该样例满足 Subtask 2 的限制。

【样例 3】

见下发文件中的 flood3.in 与 flood3.ans。

该样例满足 Subtask 3 的限制。

【样例 4】

见下发文件中的 flood4.in 与 flood4.ans。

该样例满足 Subtask 4 的限制。

【数据范围】

对所有测试数据有： $1 \leq n, m \leq 3000, 1 \leq h_i \leq 10^9, 1 \leq l_i \leq r_i \leq 10^9$ 。

子任务编号	分值	特殊限制
Subtask 1	10	$n, m \leq 5, 1 \leq l_i \leq r_i \leq 10$
Subtask 2	50	$n, m \leq 90$
Subtask 3	10	$u_i = i, v_i = i + 1$
Subtask 4	30	无特殊限制

流量 (flow)

【题目描述】

小 D 喜欢仙人掌，小 D 喜欢网络流。

小 D 现在有一颗 n 个点 m 条边的简单连通仙人掌，保证每个点至多在一个环上，每条边有一个容量。

小 D 向对仙人掌进行 q 次操作，每次操作形如 (opt, x, y) ：

- $opt = 1$ 时：小 D 把仙人掌中的第 x 条边的流量修改成了 y 。
- $opt = 2$ 时：请你告诉小 D 当前仙人掌上 $x \rightarrow y$ 的最大流。

由于小 D 太菜了，所以他请你帮他维护这棵仙人掌。

【输入格式】

从 `flow.in` 中读入数据。

第一行一个整数 T 表示该数据所属的 Subtask 编号（样例 $T = 0$ ）。

第二行三个整数 n, m, q 。

接下来 m 行每行三个整数 u, v, c ，表示一条边及其流量。

接下来 q 行每行三个整数 opt, x, y ，表示一次操作。

【输出格式】

输出到 `flow.out` 中。

对于每个 $opt = 2$ 的操作输出一行一个整数表示查询的最大流。

【样例 1 输入】

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

【样例 1 输出】

1	1
2	3

【样例 2】

见下发文件中的 flow2.in 与 flow2.ans。

该样例满足 Subtask 2 的限制。

【样例 3】

见下发文件中的 flow3.in 与 flow3.ans。

该样例满足 Subtask 3 的限制。

【样例 4】

见下发文件中的 flow4.in 与 flow4.ans。

该样例满足 Subtask 4 的限制。

【样例 5】

见下发文件中的 flow5.in 与 flow5.ans。

该样例满足 Subtask 5 的限制。

【样例 6】

见下发文件中的 flow6.in 与 flow6.ans。

该样例满足 Subtask 6 的限制。

【样例 7】

见下发文件中的 flow7.in 与 flow7.ans。

该样例满足 Subtask 7 的限制。

【数据范围】

对于所有数据有： $1 \leq n \leq 10^5, 1 \leq m, q \leq 2 \times 10^5$ ，边的容量任何时候都为 $[1, 10^9]$ 内的正整数，仙人掌无重边、无自环，保证原图连通。

子任务编号	分值	特殊限制
Subtask 1	5	$n \leq 600, q \leq 20$
Subtask 2	10	原图是一棵树
Subtask 3	10	原图是一个环
Subtask 4	15	原图是一棵基环树

子任务编号	分值	特殊限制
Subtask 5	10	边的容量始终为 1
Subtask 6	20	没有修改操作
Subtask 7	30	无特殊限制