Содержание

В некоторых задачах большой ввод и вывод. Пользуйтесь быстрым вводом-выводом.

Задача 2A. Персистентная очередь [0.45 sec, 256 mb]

Реализуйте персистентную очередь.

Формат входных данных

Первая строка содержит количество действий n ($1 \leqslant n \leqslant 200\,000$).
В строке номер $i + 1$ содержится описание действия i:

- `1 t m` — добавить в конец очереди номер t ($0 \leqslant t < i$) число m;
- `-1 t` — удалить из очереди номер t ($0 \leqslant t < i$) первый элемент.

В результате действия i, описанного в строке $i + 1$ создается очередь номер i.
Изначально имеется пустая очередь с номером ноль.
Все числа во входном файле целые, и помещаются в знаковый 32-битный тип.

Формат выходных данных

Для каждой операции удаления выведите удаленный элемент на отдельной строке.

Примеры

stdin	stdout
10 1 0 1 1 1 2 1 2 3 1 2 4 -1 3 -1 5 -1 6 -1 4 -1 8 -1 9	1 2 3 1 2 4

Задача 2B. k-я статистика на отрезке [4 sec, 256 mb]

k-ю статистику на отрезке $[l..r]$ массива A можно определить следующим способом:

```
int get( int l, int r, int k ) {
  B = отрезок [l..r] массива A
  sort(B)
  return B[k]
}
```

Дан массив. Ваша задача — много раз отвечать на запрос "k-я статистика на отрезке".

Формат входных данных

Первая строка содержит целое число N, количество чисел в массиве ($1 \leqslant N \leqslant 450\,000$).
Вторая строка используется, чтобы сгенерировать массив $a_1, a_2, \ldots, a_N$.
Она содержит три целых числа a_1, l и m ($0 \leqslant a_1, l, m < 10^9$).

$$a_i = (a_{i-1} \cdot l + m) \bmod 10^9, \quad 2 \leqslant i \leqslant N$$

Третья строка содержит целое число B — число групп запросов ($1 \leqslant B \leqslant 1000$). B следующих строк описывают группы запросов. Каждая группа описывается десятью целыми числами. Первым идет число G — количество запросов. Затем следуют x_1, l_x и m_x, потом y_1, l_y и m_y, и наконец, k_1, l_k и m_k ($1 \leqslant x_1 \leqslant y_1 \leqslant N$, $1 \leqslant k_1 \leqslant y_1 - x_1 + 1$, $0 \leqslant l_x, m_x, l_y, m_y, l_k, m_k < 10^9$). Они используются, чтобы сгенерировать вспомогательную последовательность x_g и y_g и текущие параметры i_g, j_g и k_g для $1 \leqslant g \leqslant G$:

$$\begin{array}{rcll}
x_g & = & ((i_{g-1} - 1) \cdot l_x + m_x) \bmod N) + 1, & 2 \leqslant g \leqslant G \\
y_g & = & ((j_{g-1} - 1) \cdot l_y + m_y) \bmod N) + 1, & 2 \leqslant g \leqslant G \\
i_g & = & \min(x_g, y_g), & 1 \leqslant g \leqslant G \\
j_g & = & \max(x_g, y_g), & 1 \leqslant g \leqslant G \\
k_g & = & (((k_{g-1} - 1) \cdot l_k + m_k) \bmod (j_g - i_g + 1)) + 1, & 2 \leqslant g \leqslant G
\end{array}$$

Сгенерированные параметры означают, что в g-м запросе, Нужно узнать k_g-ую статистику на отрезке $[i_g, j_g]$ массива. Общее количество запросов по всем группам не превышает 600 000.

Формат столь необычный, чтобы тесты были маленькие по объёму.

Формат выходных данных

Выведите одно число: сумму всех полученных статистик.

Пример

stdin	stdout
5 1 1 1 5 1 1 0 0 3 0 0 2 0 0 1 2 0 0 5 0 0 3 0 0 1 1 0 0 5 0 0 5 0 0 1 3 0 0 3 0 0 1 0 0 1 1 0 0 4 0 0 1 0 0	15

Замечание

Будьте аккуратны при генерации запросов. Часто ошибаются именно в этой части.

Задача 2C. Intercity Express [1.5 sec, 256 mb]

Андрей разрабатывает систему для продажи железнодорожных билетов. Он собирается протестировать ее на Междугородней Экспресс линии, которая соединяет два больших города и имеет $n-2$ промежуточных станций, то есть в итоге есть n станций, пронумерованных от 1 до n.

В Междугороднем Экспресс поезде есть s мест, пронумерованных с 1 до s. В тестирующем режиме система имеет доступ к базе данных, содержащей проданные билеты в направлении от станции 1 до станции n и должна отвечать на вопросы, можно ли продать билет от станции a до станции b, и если да, нужно найти минимальный номер места, которое свободно на протяжении всего пути между a и b.

Изначально система имеет только доступ на чтение, то есть даже если есть свободное место, она должна сообщить об этом, но не должна изменять данные.

Помогите Андрею протестировать его систему написанием программы, которые будет находить ответы на вопросы.

Формат входных данных

Первая строка содержит число n — количество станций, s — количество мест и m — количество уже проданных билетов ($2 \leqslant n \leqslant 10^9$, $1 \leqslant s \leqslant 100\,000$, $0 \leqslant m \leqslant 100\,000$).

В следующих m строках описаны билеты, описание каждого билета состоит из трех чисел: c_i, a_i и b_i — номер места, которое занимает владелец билета, номер станции, с которой продан билет и номер станции, до которой продан билет ($1 \leqslant c_i \leqslant s$, $1 \leqslant a_i < b_i \leqslant n$).

Следующие строки содержат число q — количество запросов ($1 \leqslant q \leqslant 100\,000$). Специальное значение p должно поддерживаться в течение считывания запросов. Изначально $p = 0$.

Следующие $2q$ строк описывают запросы. Каждый запрос описывается двумя числами: x_i и y_i ($x_i \leqslant y_i$).

Чтобы получить города a и b между которыми нужно проверить наличие места, используется следующая формула:

$a = x_i + p$, $b = y_i + p$. Ответ на запрос — число 0, если нет места на каждом отрезке между a и b, или минимальный номер свободного места.

После ответа на запрос, надо приравнять число p полученному ответу на запрос.

Формат выходных данных

Для каждого запроса выведите ответ на него.

Примеры

stdin	stdout
5 3 5 1 2 5 2 1 2 2 4 5 3 2 3 3 3 4 10 1 2 1 2 1 2 2 3 -2 0 2 4 1 3 1 4 2 5 1 5	1 2 2 3 0 2 0 0 0 0

Замечание

Обратите внимание, что запросы выглядят так:
$(1, 2)$, $(2, 3)$, $(3, 4)$, $(4, 5)$, $(1, 3)$, $(2, 4)$, $(3, 5)$, $(1, 4)$, $(2, 5)$, $(1, 5)$.

Задача 2D. ВнутренНЯя точка 2 [3.5 sec, 256 mb]

Дан совсем невыпуклый *простой* N-угольник и K точек. Напомним, N-угольник называется простым, если не имеет ни самопересечений, ни самокасаний. Для каждой точки нужно определить, где она находится — внутри, на границе, или снаружи.

Формат входных данных

В первой строке дано целое число T — количество тестов.
Далее идут T тестов. Тесты разделены переводом строки.
N ($3 \leqslant N \leqslant 10^5$). Далее N точек — вершины многоугольника.
K ($0 \leqslant K \leqslant 10^5$). Далее K точек — запросы.
Все координаты — целые числа по модулю не превосходящие 10^9.
Суммарное количество N и K не превосходит $2 \cdot 10^6$.

Формат выходных данных

Для каждого запроса одна строка — `INSIDE`, `BORDER` или `OUTSIDE`.
Тесты следует разделять переводом строки.

Примеры

stdin	stdout
1 4 0 0 2 0 2 2 0 2 4 1 1 0 0 0 1 0 3	INSIDE BORDER BORDER OUTSIDE