

Задача А. Инверсии на дереве

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Дано дерево из N вершин, пронумерованных от 1 до N .

Для фиксированной вершины u упорядоченная пара вершин (w, v) называется *инверсией относительно u* , если выполняются следующие условия:

- Вершина w лежит на пути от u до v (возможно, совпадает с u или v);
- Выполнено $w > v$.

Обозначим через $\mathcal{I}(u)$ количество инверсий относительно вершины u .

Для каждой вершины u найдите значение $\mathcal{I}(u)$.

Формат входных данных

В первой строке вводится одно целое число N — количество вершин дерева ($2 \leq N \leq 2 \cdot 10^5$).

В каждой из следующих $N - 1$ строк вводятся два целых числа u_i и v_i — номера вершин, соединённых i -м ребром ($1 \leq u_i, v_i \leq N$).

Гарантируется, что заданные рёбра образуют дерево.

Формат выходных данных

Выведите N целых чисел $\mathcal{I}(1), \mathcal{I}(2), \dots, \mathcal{I}(N)$.

Примеры

стандартный ввод	стандартный вывод
7 1 2 2 3 2 4 4 5 4 6 6 7	0 1 3 4 8 9 15
15 14 9 9 1 1 6 6 12 12 2 2 15 15 4 4 11 11 13 13 3 3 8 8 10 10 7 7 5	36 29 32 29 48 37 45 37 44 42 33 36 35 57 35
24 7 18 4 2 5 8 5 15 6 5 13 8 4 6 7 11 23 16 6 18 24 16 14 21 20 15 16 18 3 16 11 10 9 11 15 14 12 19 5 1 9 17 5 22 11 19	20 20 41 20 21 20 28 28 43 44 36 63 40 46 34 40 59 28

Замечание

В первом примере множества инверсий для каждой вершины имеют следующий вид:

$$u = 1 : \emptyset;$$

$$u = 2 : (2, 1);$$

$$u = 3 : (3, 1), (2, 1), (3, 2);$$

$$u = 4 : (4, 1), (2, 1), (4, 2), (4, 3);$$

$$u = 5 : (5, 1), (4, 1), (2, 1), (5, 2), (4, 2), (5, 3), (4, 3), (5, 4);$$

$$u = 6 : (6, 1), (4, 1), (2, 1), (6, 2), (4, 2), (6, 3), (4, 3), (6, 4), (6, 5);$$

$$u = 7 : (7, 1), (6, 1), (4, 1), (2, 1), (7, 2), (6, 2), (4, 2), (7, 3), \\ (6, 3), (4, 3), (7, 4), (6, 4), (7, 5), (6, 5), (7, 6).$$

Задача В. Разнообразные отрезки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 1024 мегабайта

Дан массив $a_0, a_1, \dots, a_{n-1}$. Необходимо обработать запросы, в каждом из которых требуется найти количество различных чисел на подотрезке $[l_i, r_i]$.

Запросы заданы в зашифрованной форме. Пусть p — ответ на предыдущий запрос (для первого запроса $p = 0$). Для i -го запроса даны числа x_i и y_i , а границы l_i и r_i вычисляются следующим образом:

- $l_i = \min((x_i + p) \bmod n, (y_i + p) \bmod n)$,
- $r_i = \max((x_i + p) \bmod n, (y_i + p) \bmod n)$.

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 5 \cdot 10^5$) — длина массива. Во второй строке содержатся n целых чисел $a_0, a_1, \dots, a_{n-1}$ ($0 \leq a_i \leq 10^9$).

В третьей строке дано число q ($1 \leq q \leq 5 \cdot 10^5$) — количество запросов. В каждой из следующих q строк заданы два числа x_i и y_i ($0 \leq x_i, y_i < n$).

Формат выходных данных

На каждый запрос выведите одно число — количество различных чисел на отрезке $[l_i, r_i]$.

Пример

стандартный ввод	стандартный вывод
5	3
0 1 1 0 2	1
5	1
0 4	2
3 3	3
0 1	
4 2	
0 2	

Задача С. Отсутствующий на отрезках

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

Дан массив $a_0, a_1, \dots, a_{n-1}$. Необходимо обработать запросы, в каждом из которых требуется найти МЕХ чисел на подотрезке $[l_i, r_i]$.

Запросы заданы в зашифрованной форме. Пусть p — ответ на предыдущий запрос (для первого запроса $p = 0$). Для i -го запроса даны числа x_i и y_i , а границы l_i и r_i вычисляются следующим образом:

- $l_i = \min((x_i + p) \bmod n, (y_i + p) \bmod n)$,
- $r_i = \max((x_i + p) \bmod n, (y_i + p) \bmod n)$.

Напомним, что $\text{МЕХ}(S)$ набора чисел S это наименьшее целое неотрицательное число, которого нет в S . Так, например, $\text{МЕХ}(1, 2, 3) = 0$, а $\text{МЕХ}(0, 1, 2, 4) = 3$.

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 5 \cdot 10^5$) — длина массива. Во второй строке содержатся n целых чисел $a_0, a_1, \dots, a_{n-1}$ ($0 \leq a_i \leq 10^9$).

В третьей строке дано число q ($1 \leq q \leq 5 \cdot 10^5$) — количество запросов. В каждой из следующих q строк заданы два числа x_i и y_i ($0 \leq x_i, y_i < n$).

Формат выходных данных

На каждый запрос выведите одно число — МЕХ чисел на отрезке $[l_i, r_i]$.

Пример

стандартный ввод	стандартный вывод
5	3
0 1 1 0 2	0
5	1
0 4	3
3 4	2
3 4	
1 3	
3 0	

Замечание

Реальные отрезки запросов в первом примере это $[0, 4]$, $[1, 2]$, $[3, 4]$, $[2, 4]$, $[1, 3]$ и ответы на них это $\text{МЕХ}(0, 1, 1, 0, 2) = 3$, $\text{МЕХ}(1, 1) = 0$, $\text{МЕХ}(0, 2) = 1$, $\text{МЕХ}(1, 0, 2) = 3$, $\text{МЕХ}(1, 1, 0) = 2$

Задача D. Порядковая статистика на дереве

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Дано дерево из n вершин, пронумерованных числами от 1 до n . Рёбра дерева занумерованы числами от 1 до $n - 1$ так, что каждое число встречается ровно на одном ребре (то есть номера рёбер образуют перестановку чисел от 1 до $n - 1$).

Для пары различных вершин u и v рассмотрим простой путь между ними — последовательность рёбер, которую нужно пройти, чтобы попасть из u в v . k -й порядковой статистикой на этом пути называется число, которое окажется на k -м месте, если выписать номера всех рёбер пути в порядке необывания.

Вам нужно ответить на q запросов вида (u, v, k) : для пути между вершинами u и v найти k -ю порядковую статистику среди номеров рёбер, лежащих на этом пути.

Формат входных данных

В первой строке записаны два целых числа n и q ($2 \leq n \leq 200\,000$; $1 \leq q \leq 300\,000$) — количество вершин дерева и количество запросов.

В следующих $n - 1$ строках описаны рёбра дерева: i -я из этих строк содержит три целых числа u_i , v_i и w_i ($1 \leq u_i, v_i \leq n$; $u_i \neq v_i$; $1 \leq w_i \leq n - 1$) — номера вершин, которые соединяет i -е ребро, и номер этого ребра. Гарантируется, что данные рёбра образуют дерево, а числа w_i различны (то есть образуют перестановку чисел от 1 до $n - 1$).

В следующих q строках описаны запросы: j -я из этих строк содержит три целых числа u , v и k ($1 \leq u, v \leq n$; $u \neq v$) — вершины, задающие путь, и номер порядковой статистики. Гарантируется, что k не превосходит количества рёбер на пути между u и v .

Формат выходных данных

Выведите q строк — ответы на запросы в том порядке, в котором они заданы во входных данных.

Примеры

стандартный ввод	стандартный вывод
5 3 1 2 3 1 3 1 3 4 4 3 5 2 2 5 1 2 4 3 4 5 2	1 4 4
4 2 1 2 2 2 3 3 3 4 1 1 4 2 1 3 1	2 2

Замечание

В первом примере дерево выглядит так: вершина 1 соединена с вершиной 2 ребром номер 3 и с вершиной 3 ребром номер 1; вершина 3, в свою очередь, соединена с вершиной 4 ребром номер 4 и с вершиной 5 ребром номер 2.

- Путь из 2 в 5 проходит по рёбрам $2 \rightarrow 1 \rightarrow 3 \rightarrow 5$ с номерами 3, 1, 2. В отсортированном виде: 1, 2, 3. Первая порядковая статистика равна 1.

- Путь из 2 в 4 проходит по рёбрам $2 \rightarrow 1 \rightarrow 3 \rightarrow 4$ с номерами 3, 1, 4. В отсортированном виде: 1, 3, 4. Третья порядковая статистика (то есть наибольшее число) равна 4.
- Путь из 4 в 5 проходит по рёбрам $4 \rightarrow 3 \rightarrow 5$ с номерами 4, 2. В отсортированном виде: 2, 4. Вторая порядковая статистика равна 4.

Во втором примере дерево является цепочкой $1-2-3-4$ с номерами рёбер 2, 3, 1 соответственно. Путь из 1 в 4 содержит все рёбра с номерами 2, 3, 1, отсортированные — 1, 2, 3; вторая порядковая статистика равна 2. Путь из 1 в 3 содержит рёбра с номерами 2, 3; первая (то есть наименьшая) порядковая статистика равна 2.

Задача E. Range Min Count Queries

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Дан массив $A_0, A_1, \dots, A_{n-1}$.

Ответом на запрос $[l, r]$ называется количество вхождений минимального значения на этом отрезке. Иными словами, ответ равен количеству индексов i таких, что $l \leq i \leq r$ и $A_i = \min(A_l, A_{l+1}, \dots, A_r)$.

Запросы заданы в сжатом виде и разбиты на блоки. Каждый блок задаётся пятью числами k, l, r, a и b и содержит k запросов, которые могут быть вычислены по формулам.

- Положим $x_1 = l, y_1 = r$.
- Для $j \in [1, k]$ положим $L_j = \min(x_j, y_j), R_j = \max(x_j, y_j)$.
Запрос под номером j задаётся отрезком $[L_j, R_j]$. Обозначим ответ на него ans_j .
- Для $j < k$ можно вычислить x_{j+1} и y_{j+1} по x_j, y_j, ans_j .

$$x_{j+1} = (a \cdot x_j + \text{ans}_j) \bmod n,$$

$$y_{j+1} = (b \cdot y_j + \text{ans}_j) \bmod n.$$

Для каждого блока найдите сумму ответов на все запросы в нём: $\text{ans}_1 + \text{ans}_2 + \dots + \text{ans}_k$.

Формат входных данных

В первой строке вводится целое число n — длина массива ($1 \leq n \leq 10^6$).

Во второй строке вводятся n целых чисел $A_0, A_1, \dots, A_{n-1}$ ($-10^9 \leq A_i \leq 10^9$).

В следующей строке вводится целое число q — количество блоков запросов ($1 \leq q \leq 10^5$).

В каждой из следующих q строк вводятся пять целых чисел k_i, l_i, r_i, a_i и b_i — параметры i -го блока ($1 \leq k_i \leq 10^7, 0 \leq l_i, r_i, a_i, b_i < n$).

Гарантируется, что $\sum_{i=1}^q k_i \leq 10^7$.

Формат выходных данных

Для каждого блока запросов выведите в отдельной строке одно целое число — сумму ответов на все запросы этого блока.

Примеры

стандартный ввод	стандартный вывод
6 2 1 1 3 1 4 3 1 0 5 0 0 1 1 2 0 0 1 3 5 0 0	3 2 1
8 3 1 1 2 1 4 1 1 2 2 0 1 0 3 3 0 2 1 2	4 7
6 1 1 2 2 1 5 1 67 0 1 3 2	90

Замечание

Во втором примере в первом блоке выполняются два запроса: $[0, 1]$, $[1, 4]$. Во втором блоке выполняются три запроса: $[0, 2]$, $[2, 6]$, $[5, 7]$.

Задача F. Операции с таблицей

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам дана таблица $n \times m$, в каждой клетке которой записано число 0. Вам требуется ответить на q запросов трех типов:

- 1 $l\ r\ x$. Добавить число x ко всем числам, лежащим в столбцах $l, l+1, \dots, r$;
- 2 $i\ x$. Заменить все элементы в i -й строке на x ;
- 3 $i\ j$. Вывести значение элемента из i -й строки, j -го столбца.

Формат входных данных

В первой строке указана тройка n, m и q ($1 \leq n, m, q \leq 200\,000$) — размеры таблицы и количество запросов.

Затем следуют описания запросов в следующем формате:

- 1 $p\ q\ y$. ($1 \leq p, q \leq m, 1 \leq y \leq 10^9$)
- 2 $s\ y$. ($1 \leq i \leq n, 1 \leq y \leq 10^9$)
- 3 $s\ c$. ($1 \leq i \leq n, 1 \leq j \leq m$)

Обратите внимание, что запросы зашифрованы! Чтобы их расшифровать, требуется поддерживать переменную α — ответ на последний запрос третьего типа, изначально $\alpha = 0$.

Значения x, i, j, l, r для всех запросов восстанавливаются по формуле:

$$x = 1 + ((\alpha + y) \bmod 10^9)$$

$$i = 1 + ((\alpha + s) \bmod n)$$

$$j = 1 + ((\alpha + c) \bmod m)$$

$$l = 1 + ((\alpha + p) \bmod m)$$

$$r = 1 + ((\alpha + q) \bmod m)$$

Гарантируется, что $l \leq r$ для всех запросов первого типа.

Формат выходных данных

Для каждого запроса третьего типа выведите ответ на него в отдельной строке.

Пример

стандартный ввод	стандартный вывод
3 3 9	1
1 3 1 1000000000	2
3 1 1	2
2 1 1000000000	5
3 1 1	3
3 3 1	4
1 2 3 1000000000	
3 3 2	
3 2 3	
3 3 1	

Задача G. Порядковая статистика на дереве отрезков

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 1024 мегабайта

Дано дерево из n вершин, пронумерованных числами от 1 до n . На каждом ребре дерева записана пара целых чисел $[l, r]$ ($0 \leq l \leq r \leq n - 1$), и считается, что на этом ребре ровно по одному разу записано каждое целое число отрезка $[l, r]$ — то есть ребро задаёт не одно число, а сразу $r - l + 1$ чисел: $l, l + 1, \dots, r$.

Рассмотрим простой путь между парой различных вершин u и v . Выпишем в один список числа со **всех** рёбер этого пути (если число встречается на нескольких рёбрах пути, оно попадает в список столько раз, на скольких рёбрах встретилось) и отсортируем список по неубыванию. k -й порядковой статистикой на пути называется число, стоящее в этом списке на k -м месте.

Вам нужно ответить на q запросов вида (u, v, k) : для пути между вершинами u и v найти k -ю порядковую статистику среди чисел, записанных на рёбрах этого пути.

Формат входных данных

В первой строке записаны два целых числа n и q ($2 \leq n \leq 200\,000$; $1 \leq q \leq 300\,000$) — количество вершин дерева и количество запросов.

В следующих $n - 1$ строках описаны рёбра дерева: i -я из этих строк содержит четыре целых числа u_i, v_i, l_i и r_i ($1 \leq u_i, v_i \leq n$; $u_i \neq v_i$; $0 \leq l_i \leq r_i \leq n - 1$) — номера вершин, которые соединяет i -е ребро, и границы отрезка чисел, записанных на этом ребре. Гарантируется, что данные рёбра образуют дерево.

В следующих q строках описаны запросы: j -я из этих строк содержит три целых числа u, v и k ($1 \leq u, v \leq n$; $u \neq v$; $k \geq 1$) — вершины, задающие путь, и номер порядковой статистики. Гарантируется, что k не превосходит суммарного количества чисел, записанных на рёбрах пути между u и v (с учётом повторений); это количество может достигать $(n - 1) \cdot n < 4 \cdot 10^{10}$, поэтому для хранения k рекомендуется использовать 64-битный целочисленный тип.

Формат выходных данных

Выведите q строк — ответы на запросы в том порядке, в котором они заданы во входных данных.

Примеры

стандартный ввод	стандартный вывод
4 3 1 2 0 2 2 3 1 1 3 4 2 3 1 4 1 1 4 6 1 3 2	0 3 1
5 3 1 2 0 1 1 3 1 2 3 4 0 0 3 5 2 3 2 4 3 2 4 5 4 5 1	1 2 0

Замечание

В первом примере дерево является цепочкой $1 - 2 - 3 - 4$. На ребре $(1, 2)$ записаны числа отрезка $[0, 2]$, то есть $0, 1, 2$; на ребре $(2, 3)$ — отрезка $[1, 1]$, то есть только число 1 ; на ребре $(3, 4)$ — отрезка

$[2, 3]$, то есть 2, 3.

- Путь из 1 в 4 содержит числа со всех трёх рёбер: 0, 1, 2, 1, 2, 3. Отсортированный список: 0, 1, 1, 2, 2, 3. Первая (наименьшая) порядковая статистика равна 0, а шестая (наибольшая, всего чисел на пути как раз шесть) равна 3.
- Путь из 1 в 3 содержит числа с рёбер (1, 2) и (2, 3): 0, 1, 2 и 1. Отсортированный список: 0, 1, 1, 2. Вторая порядковая статистика равна 1.

Во втором примере вершина 1 соединена с вершиной 2 (числа 0, 1) и с вершиной 3 (числа 1, 2); вершина 3 соединена с вершиной 4 (число 0) и с вершиной 5 (числа 2, 3).

- Путь из 2 в 4 проходит по рёбрам (2, 1), (1, 3), (3, 4) с числами 0, 1, затем 1, 2, затем 0. Отсортированный список: 0, 0, 1, 1, 2. Третья порядковая статистика равна 1, а пятая (наибольшая) равна 2.
- Путь из 4 в 5 проходит по рёбрам (4, 3) и (3, 5) с числами 0 и 2, 3. Отсортированный список: 0, 2, 3. Первая (наименьшая) порядковая статистика равна 0.

Задача Н. Массив и еще массив

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

У Ани есть перестановка $P_1, P_2, \dots, P_N$. Она обозначила её как массив $A_0 = (P_1, P_2, \dots, P_N)$ и решила с помощью Q операций получить ещё Q массивов.

На i -й операции заданы три числа (t_i, s_i, x_i) . Новый массив A_i строится из уже существующего массива A_{s_i} следующим образом:

1. Если $t_i = 1$, Аня удаляет из A_{s_i} все элементы, кроме первых x_i , и записывает удалённые элементы в массив A_i , сохраняя их порядок.
Если в A_{s_i} меньше x_i элементов, то массив A_{s_i} не изменяется, а A_i становится пустым. В частности, если $x_i = 0$, все элементы A_{s_i} переходят в A_i .
2. Если $t_i = 2$, Аня удаляет из A_{s_i} все элементы, значения которых больше x_i , и записывает их в массив A_i , сохраняя их порядок.

После каждой операции определите размер созданного массива A_i .

Формат входных данных

В первой строке вводится одно целое число N — длина исходной перестановки ($1 \leq N \leq 2 \cdot 10^5$).

Во второй строке вводятся N целых чисел $P_1, \dots, P_N$ — исходная перестановка ($1 \leq P_i \leq N$).
Гарантируется, что все P_i различны.

В третьей строке вводится одно целое число Q — количество операций ($1 \leq Q \leq 2 \cdot 10^5$).

В i -й из следующих Q строк вводятся три целых числа t_i, s_i и x_i — тип операции, номер массива, над которым выполняется операция, и её параметр соответственно ($1 \leq t_i \leq 2, 0 \leq s_i < i, 0 \leq x_i \leq N$).

Формат выходных данных

Выведите Q строк. В i -й строке выведите размер массива A_i сразу после i -й операции.

Примеры

стандартный ввод	стандартный вывод
10 1 8 7 4 5 6 3 2 9 10 5 2 0 4 1 1 2 2 0 2 2 2 5 1 0 1	6 4 2 3 1
8 6 7 8 4 5 1 3 2 5 2 0 0 1 1 0 2 2 0 1 3 8 2 2 3	8 8 8 0 0
30 20 6 13 11 29 30 9 10 16 5 8 25 1 19 12 1 18 7 2 4 27 3 22 23 24 28 21 14 26 15 17 8 10 1 0 22 1 0 21 2 0 15 1 0 9 1 3 6 2 3 18 1 6 2 1 0 1 2 5 20 2 7 26	8 1 7 4 2 5 3 8 1 1

Замечание

В примере изначально

$$A_0 = (1, 8, 7, 4, 5, 6, 3, 2, 9, 10).$$

После первой операции из A_0 удаляются все элементы, большие 4. Поэтому

$$A_0 = (1, 4, 3, 2), \quad A_1 = (8, 7, 5, 6, 9, 10).$$

После второй операции из A_1 удаляются все элементы после первых двух:

$$A_0 = (1, 4, 3, 2), \quad A_1 = (8, 7), \quad A_2 = (5, 6, 9, 10).$$

После следующих операций массивы имеют вид:

$$\begin{aligned} A_0 &= (1, 2), \quad A_1 = (8, 7), \quad A_2 = (5, 6, 9, 10), \quad A_3 = (4, 3); \\ A_0 &= (1, 2), \quad A_1 = (8, 7), \quad A_2 = (5), \quad A_3 = (4, 3), \quad A_4 = (6, 9, 10); \\ A_0 &= (1), \quad A_1 = (8, 7), \quad A_2 = (5), \quad A_3 = (4, 3), \quad A_4 = (6, 9, 10), \quad A_5 = (2). \end{aligned}$$

Задача I. King and ICPC

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 1024 мегабайта

Тёмные времена нависли над АСМ-ом. Тысячи команд АСМ-щиков вышли на защиту от нападения СТФ-щиков. Всего имеется N команд, пронумерованных от 1 до N . Каждая команда, по традиции, состоит из трёх человек. Каждому человеку присвоили параметр АСМ-овости, выраженный целым числом.

Король АСМ-щиков хочет создать армию, выбрав из каждой команды ровно по одному человеку так, чтобы суммарная АСМ-овость была как можно больше, но при этом делилась нацело на число D .

Так как АСМ-щики по своей натуре любят усложнять задачи, они решили рассмотреть несколько отрезков $[l_i, r_i]$, составляя армию лишь из команд с номерами из этого отрезка.

Король поручил вам обработку этих запросов. На горизонте уже видны флаги, у вас осталось меньше пяти часов!

Формат входных данных

Первая строка содержит два целых числа N ($1 \leq N \leq 50\,000$) и D ($1 \leq D \leq 50$).

В следующих N строках содержится описание каждой команды в виде трёх целых чисел. АСМ-овость каждого человека является целым числом от 0 до 10^9 .

Следующая строка содержит целое число M — количество отрезков ($1 \leq M \leq 300\,000$).

В следующих M строках идёт описание каждого запроса в виде пары чисел l_i и r_i ($1 \leq l_i \leq r_i \leq N$).

Формат выходных данных

На каждый отрезок выведите в отдельной строке максимальную сумму, которую можно на нём собрать. Если же сумму, делящуюся на D собрать невозможно, выведите «-1» (без кавычек).

Пример

стандартный ввод	стандартный вывод
2 2	0
0 1 3	2
1 2 3	6
3	
1 1	
2 2	
1 2	

Задача J. Комбокамень

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Одна большая компания разрабатывает компьютерную игру «Комбокамень», которая должна мигом перевернуть всю индустрию. Правила игры достаточно сложные, и на реализацию серверного движка, моделирующего ход игры, был объявлен конкурс. От вас требуется реализовать подобный серверный движок.

Суть игры — игроки умеют призывать на арену и усиливать существ, используя заклинания, а также заставлять их сражаться друг с другом. Каждое существо имеет два параметра — численное значение атаки a и численное значение оставшегося здоровья h . Для краткости будем обозначать параметры существа как (a, h) . Исходно на арене нет существ.

Игроку доступны следующие заклинания:

- *Призыв существа*: Призвать новое существо с характеристиками $(1, 1)$. Если уже в игру было введено k существ, то новое существо получает номер $k + 1$.
- *Благословение силы*: Удвоить атаку выбранного существа. Если до применения этого заклинания оно имело характеристики (a, h) , то после этого действия оно будет иметь характеристики $(2a, h)$.
- *Божественный дух*: Удвоить здоровье выбранного существа. Если до применения этого заклинания оно имело характеристики (a, h) , то после этого действия оно будет иметь характеристики $(a, 2h)$.
- *Копия из лавы*: Призвать новое существо, которое будет иметь такие же характеристики, как и выбранное заклинанием существо. Если уже в игру было введено k существ, то новое существо получает номер $k + 1$.
- *Сражайся!*: Заставить двух различных существ сразиться. Во время сражения оба существа одновременно наносят друг другу по одному удару, уменьшая количество здоровья соперника на значение своей атаки. Так, если сражаются два существа с характеристиками (a_1, h_1) и (a_2, h_2) , то после сражения они будут иметь характеристики $(a_1, h_1 - a_2)$ и $(a_2, h_2 - a_1)$, соответственно. Если после сражения у существа остается 0 или меньше единиц здоровья, оно умирает и больше не может участвовать в игре.

От серверного движка, на реализацию которого объявлен конкурс, требуется способность про-моделировать все события и для каждого созданного во время игры существа вывести номер хода, на котором оно погибло, либо определить, что оно осталось живо к концу игры.

Кроме того, движок должен корректно обрабатывать случаи, когда игрок пытается взаимодействовать с мертвыми по мнению сервера существами: если заклинание *Благословение силы*, *Божественный дух* или *Сражайся!* обращено к уже мертвому существу, то не должно произойти ничего. Если заклинание *Копия из лавы* применено к мертвому существу, создается его мертвая копия с такими же характеристиками, но умершая на текущем ходу, в момент копирования.

Формат входных данных

В первой строке дано число n — количество совершенных ходов ($1 \leq n \leq 250\,000$).

В следующих n строках даны ходы, пришедшие к серверному движку, в следующем формате:

- 1 — применить заклинание *Призыв существа*;
- 2 i — применить заклинание *Благословение силы* к существу с номером i ;
- 3 i — применить заклинание *Божественный дух* к существу с номером i ;
- 4 i — применить заклинание *Копия из лавы* к существу с номером i ;
- 5 $i\ j$ — применить заклинание *Сражайся!* к существам с номерами i и j .

Гарантируется, что любые упомянутые в запросах существа к моменту запроса уже были призваны, но, возможно, могут уже быть мертвы.

Формат выходных данных

В первой строке выведите одно целое число k — количество существ, призванных за время игры.

В следующей строке выведите k целых чисел $t_1, t_2, \dots, t_k$ — если существо с номером i осталось живо к концу игры, то t_i должно быть равно -1 , иначе t_i должно быть равно номеру хода, на котором оно погибло.

Пример

стандартный ввод	стандартный вывод
16	5
1	13 5 14 -1 16
2 1	
3 1	
1	
5 1 2	
3 1	
1	
3 3	
3 3	
4 1	
5 1 3	
3 3	
5 1 3	
5 4 3	
5 4 3	
4 1	

Замечание

В таблице можно увидеть, как изменялись характеристики существ в первом примере.

ход	1	2	3	4	5
0	-	-	-	-	-
1	(1, 1)	-	-	-	-
2	(2, 1)	-	-	-	-
3	(2, 2)	-	-	-	-
4	(2, 2)	(1, 1)	-	-	-
5	(2, 1)	мертво	-	-	-
6	(2, 2)	мертво	-	-	-
7	(2, 2)	мертво	(1, 1)	-	-
8	(2, 2)	мертво	(1, 2)	-	-
9	(2, 2)	мертво	(1, 4)	-	-
10	(2, 2)	мертво	(1, 4)	(2, 2)	-
11	(2, 1)	мертво	(1, 2)	(2, 2)	-
12	(2, 1)	мертво	(1, 4)	(2, 2)	-
13	мертво	мертво	(1, 2)	(2, 2)	-
14	мертво	мертво	мертво	(2, 1)	-
15	мертво	мертво	мертво	(2, 1)	-
16	мертво	мертво	мертво	(2, 1)	мертво

Задача К. Настрой в ушах

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4.5 секунд
Ограничение по памяти:	512 мегабайт

С распространением музыкальных стриминговых сервисов сильно изменилось то, как слушатели потребляют музыку. Пользователям доступны огромные каталоги треков, и меломан Илья собрал в своём аккаунте целых N любимых композиций, пронумерованных от 1 до N .

Сейчас у Ильи особенное настроение, и для каждого трека i он знает число A_i *соответствие* этого трека состоянию Ильи. Все значения A_i целые и лежат в диапазоне от 1 до K .

Когда Илья выбирает отрезок треков $[l, r]$ и начинает их слушать, он может пропускать некоторые из них. Для хорошего опыта важно, чтобы последовательность значений A_i у прослушанных треков не убывала. Найдите число способов прослушать треки так, чтобы опыт прослушивания получился хорошим.

Иными словами, от вас требуется найти количество способов выбрать непустую подпоследовательность треков на отрезке $[l, r]$, такую что их соответствия A_i образуют неубывающую последовательность. Поскольку ответ может быть очень большим, выведите его по модулю 998 244 353.

Формат входных данных

В первой строке заданы два целых числа N и K ($1 \leq N \leq 2 \cdot 10^5$, $1 \leq K \leq 100$) — количество треков и диапазон возможных значений.

Во второй строке содержатся N целых чисел $A_1, A_2, \dots, A_N$ ($1 \leq A_i \leq K$) — соответствия треков настроению.

В третьей строке дано целое число Q ($1 \leq Q \leq 4 \cdot 10^5$) — количество запросов.

В каждой из следующих Q строк даны два целых числа l и r ($1 \leq l \leq r \leq N$) — границы отрезка запроса.

Формат выходных данных

Для каждого запроса выведите одно число — количество неубывающих непустых подпоследовательностей на отрезке $[l, r]$ по модулю 998 244 353.

Примеры

стандартный ввод	стандартный вывод
5 3 2 1 2 3 3 4 1 3 2 4 1 5 3 3	5 7 23 1
10 5 4 2 3 3 3 4 2 1 2 1 8 7 8 8 9 10 10 6 9 9 10 1 7 5 6 9 9	2 3 1 6 2 35 3 1

Замечание

В первом примере неубывающие последовательности это

1. В первом запросе: $(2), (1), (2), (2, 2), (1, 2)$.
2. Во втором запросе: $(1), (2), (3), (1, 2), (1, 3), (2, 3), (1, 2, 3)$.
3. В четвёртом запросе:
 $(2), (1), (2), (3), (3), (2, 2), (2, 3), (2, 3), (1, 2), (1, 3),$
 $(1, 3), (2, 3), (2, 3), (3, 3), (2, 2, 3), (2, 2, 3), (2, 3, 3),$
 $(1, 2, 3), (1, 2, 3), (1, 3, 3), (2, 3, 3), (2, 2, 3, 3), (1, 2, 3, 3)$
4. В пятом запросе: (2) .

Задача L. Изменяемое разнообразие

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 512 мегабайт

Дан массив $A_1, A_2, \dots, A_N$. Необходимо обработать запросы двух видов:

- $1\ l\ r$ — найти количество различных чисел среди $A_l, A_{l+1} \dots A_r$;
- $2\ i\ x$ — обновить значение $A_i = x$.

Формат входных данных

В первой строке дано целое число N ($1 \leq N \leq 10^5$) — длина массива. Во второй строке содержатся N целых чисел $A_1, A_2, \dots, A_N$ ($0 \leq A_i \leq 10^9$).

В третьей строке дано число q ($1 \leq q \leq 10^5$) — количество запросов. В каждой из следующих q строк заданы запросы в одном из двух форматов:

- $1\ l\ r$ ($1 \leq l \leq r \leq n$);
- $2\ i\ x$ ($1 \leq i \leq n, 0 \leq x \leq 10^9$).

Формат выходных данных

В ответ на каждый запрос первого типа выведите одно число — количество различных чисел на отрезке.

Пример

стандартный ввод	стандартный вывод
5	3
0 1 1 0 2	2
8	3
1 1 5	2
1 2 4	4
1 2 5	1
2 4 2	
1 2 5	
2 2 3	
1 1 5	
1 4 4	