

Задача А. Максимумы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 6 секунд
Ограничение по памяти: 256 мегабайт

К сожалению, у Дани не хватило времени, чтобы написать нескучное условие к этой задаче.

Вам дан массив a_0, \dots, a_{n-1} , который задаётся числами n, a_0, b, c, d следующим образом:

$$b_i = (b_{i-1} \cdot c + d) \bmod 2^{31} \text{ для } i \geq 1$$

$$a_i = (a_{i-1} + 1 - 2 \cdot ((b_i \bmod 239179) \bmod 2)) \text{ для } i \geq 1$$

Обратите внимание, что два соседних числа отличаются либо на $+1$, либо на -1 .

Ответьте на n запросов, i -й запрос — максимум на отрезке $[\min(l_i, r_i), \max(l_i, r_i)]$ для $i = 0 \dots n-1$.

Пусть ans_i — ответ на i -й запрос. Будем считать, что $ans_{-1} = 0$. Вам задаётся число x_0 . Далее, l_i и r_i, x_i вычисляются так:

$$l_i = (x_i + ans_{i-1}) \bmod n \text{ для } i \geq 0$$

$$r_i = (l_i + i) \bmod n \text{ для } i \geq 0$$

$$x_i = (x_{i-1} \cdot 1103515245 + 12345) \bmod 2^{31} \text{ для } i \geq 1$$

При взятии по модулю обратите внимание на то, что ans_i бывают отрицательными.

Формат входных данных

Единственная строка ввода содержит шесть целых чисел n, a_0, b, c, d, x_0 .

- $1 \leq n \leq 2 \cdot 10^7$
- $-10^9 \leq a_0 \leq 10^9$
- $0 \leq b, c, d, x_0 \leq 2^{31} - 1$

Формат выходных данных

Выведите сумму ответов на все запросы.

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4 0	11
100500 -1 23 45 67 89	-8614564

Задача В. Максимумы возвращаются

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 6 секунд
Ограничение по памяти: 512 мегабайт

Вам дан массив a_0, \dots, a_{n-1} , который задаётся числами n, a_0, b_0, c, d следующим образом:

$$b_i = (b_{i-1} \cdot c + d) \bmod 2^{31} \text{ для } i \geq 1$$

$$a_i = (a_{i-1} + 1 - 2 \cdot ((b_i \bmod 239179) \bmod 2)) \text{ для } i \geq 1$$

Обратите внимание, что два соседних числа отличаются либо на $+1$, либо на -1 .

Ответьте на n запросов, i -й запрос — поиск количества максимумов на отрезке $[\min(l_i, r_i), \max(l_i, r_i)]$ для $i = 0 \dots n - 1$. Пусть ans_i — ответ на i -й запрос. Будем считать, что $ans_{-1} = 0$. Вам задаётся число x_0 . Далее, l_i и r_i, x_i вычисляются так:

$$l_i = (x_i + ans_{i-1}) \bmod n \text{ для } i \geq 0$$

$$r_i = (l + i) \bmod n \text{ для } i \geq 0$$

$$x_i = (x_{i-1} \cdot 1103515245 + 12345) \bmod 2^{31} \text{ для } i \geq 1$$

Формат входных данных

Единственная строка ввода содержит шесть целых чисел n, a_0, b, c, d, x_0 .

- $1 \leq n \leq 2 \cdot 10^7$
- $-10^9 \leq a_0 \leq 10^9$
- $0 \leq b, c, d, x_0 \leq 2^{31} - 1$

Формат выходных данных

Выведите сумму ответов на все запросы.

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4 0	4
100500 -1 23 45 67 89	173287

Задача С. Соединение и разъединение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Вы когда-нибудь слышали про обход в глубину? Например, используя этот алгоритм, вы можете проверить является ли граф связным за время $O(E)$. Вы можете даже посчитать количество компонент связности за то же время.

А вы когда-нибудь слышали про систему непересекающихся множеств? Используя эту структуру, вы можете быстро обрабатывать запросы “Добавить ребро в граф” и “Посчитать количество компонент связности в графе”.

А вы когда-нибудь слышали о *динамической* задаче связности? В этой задаче вам необходимо обрабатывать три типа запросов:

1. Добавить ребро в граф.
2. Удалить ребро из графа.
3. Посчитать количество компонент связности в графе.

Можно считать, что граф является неориентированным. Изначально граф является пустым.

Формат входных данных

В первой строке находятся два целых числа N и K — количество вершин и количество запросов, соответственно ($1 \leq N \leq 300\,000$, $0 \leq K \leq 300\,000$). Следующие K строк содержат запросы, по одному в строке. Каждый запрос имеет один из трех типов:

1. $+ u v$: Добавить ребро между вершинами u и v . Гарантируется, что такого ребра нет.
2. $- u v$: Удалить ребро между u и v . Гарантируется, что такое ребро есть.
3. $?$: Посчитать количество компонент связности в графе.

Вершины пронумерованы целыми числами от 1 до N . Во всех запросах $u \neq v$.

Формат выходных данных

Для каждого запроса типа ‘?’, Выведите количество компонент связности в момент запроса.

Пример

стандартный ввод	стандартный вывод
5 11	5
?	1
+ 1 2	1
+ 2 3	2
+ 3 4	
+ 4 5	
+ 5 1	
?	
- 2 3	
?	
- 4 5	
?	

Задача D. Урны и шары

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Пусть у вас есть n урн, в каждой из которых лежит по одному шару. Урна с номером i содержит шарик под номером i . У вас есть специальное устройство, которое позволяет перемещать шарики. Им чрезвычайно просто пользоваться: сначала вы выбираете некоторый отрезок последовательных урн. После этого вы выбираете некоторый другой отрезок последовательных урн такой же длины, как и исходный, и затем шарики из урн первого отрезка перемещаются в соответствующие урны второго отрезка.

Дана последовательность перемещений. Установите, в какой урне окажется каждый шарик.

Формат входных данных

Первая строка входных данных содержит два числа n и m — число урн и число перемещений, соответственно ($1 \leq n \leq 100\,000$, $1 \leq m \leq 50\,000$). Каждая из следующих m строк содержит три числа $count_i$, $from_i$ и to_i , которые означают одновременное перемещение всех шариков из урны $from_i$ в урну to_i , всех шариков из урны $from_i + 1$ в урну $to_i + 1$, ..., всех шариков из урны $from_i + count_i - 1$ в урну $to_i + count_i - 1$ ($1 \leq count_i, from_i, to_i \leq n$, $\max(from_i, to_i) + count_i \leq n + 1$).

Формат выходных данных

Выведите n чисел — итоговые позиции каждого шарика.

Пример

стандартный ввод	стандартный вывод
2 3	1 1
1 1 2	
1 2 1	
1 2 1	

Задача Е. K -я порядковая статистика на отрезке

Имя входного файла: стандартный ввод
 Имя выходного файла: стандартный вывод
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 256 мегабайт

Дан массив из N неотрицательных чисел, строго меньших 10^9 . Вам необходимо ответить на несколько запросов о величине k -й порядковой статистики на отрезке $[l, r]$.

Формат входных данных

Первая строка содержит число N ($1 \leq N \leq 450\,000$) — размер массива.

Вторая строка может быть использована для генерации a_i — начальных значений элементов массива. Она содержит три числа a_1, l и m ($0 \leq a_1, l, m < 10^9$); для i от 2 до N

$$a_i = (a_{i-1} \cdot l + m) \bmod 10^9.$$

В частности, $0 \leq a_i < 10^9$.

Третья строка содержит одно целое число B ($1 \leq B \leq 1000$) — количество групп запросов.

Следующие B строк описывают одну группу запросов. Каждая группа запросов описывается 10 числами. Первое число G обозначает количество запросов в группе. Далее следуют числа x_1, l_x и m_x , затем y_1, l_y и m_y , затем, k_1, l_k и m_k ($1 \leq x_1 \leq y_1 \leq N, 1 \leq k_1 \leq y_1 - x_1 + 1, 0 \leq l_x, m_x, l_y, m_y, l_k, m_k < 10^9$). Эти числа используются для генерации вспомогательных последовательностей x_g и y_g , а также параметров запросов i_g, j_g и k_g ($1 \leq g \leq G$)

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

Сгенерированные последовательности описывают запросы, g -й запрос состоит в поиске k_g -го по величине числа среди элементов отрезка $[i_g, j_g]$.

Суммарное количество запросов не превосходит 600 000.

Формат выходных данных

Выведите единственное число — сумму ответов на запросы.

Пример

стандартный ввод	стандартный вывод
5	15
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	

Задача F. Intercity Express

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

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 ($2 \leq n \leq 10^9$, $1 \leq s \leq 100\,000$, $0 \leq m \leq 100\,000$).

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 ($1 \leq c_i \leq s$, $1 \leq a_i < b_i \leq n$).

Следующие строки содержат число q — количество запросов ($1 \leq q \leq 100\,000$). 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 ($x_i \leq y_i$).

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.

Пример

стандартный ввод	стандартный вывод
5 3 5	1
1 2 5	2
2 1 2	2
2 4 5	3
3 2 3	0
3 3 4	2
10	0
1 2	0
1 2	0
1 2	0
2 3	
-2 0	
2 4	
1 3	
1 4	
2 5	
1 5	

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 G. Снеговика

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

Зима. 2012 год. На фоне грядущего Апокалипсиса и конца света незамеченной прошла новость об очередном прорыве в областях клонирования и снеговиков: клонирования снеговиков. Вы конечно знаете, но мы вам напомним, что снеговик состоит из нуля или более вертикально поставленных друг на друга шаров, а клонирование — это процесс создания идентичной копии (клона).

В местечке Местячково учитель Андрей Сергеевич Учитель купил через интернет-магазин «Интернет-магазин аппаратов клонирования» аппарат для клонирования снеговиков. Теперь дети могут играть и даже играют во дворе в следующую игру. Время от времени один из них выбирает понравившегося снеговика, клонирует его и:

- либо добавляет ему сверху один шар;
- либо удаляет из него верхний шар (если снеговик не пустой).

Учитель Андрей Сергеевич Учитель записал последовательность действий и теперь хочет узнать суммарную массу всех построенных снеговиков.

Формат входных данных

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

- $t\ m$ — клонировать снеговика номер t ($0 \leq t < i$) и добавить сверху шар массой m ($0 < m \leq 1000$);
- $t\ 0$ — клонировать снеговика номер t ($0 \leq t < i$) и удалить верхний шар. Гарантируется, что снеговик t не пустой.

В результате действия i , описанного в строке $i + 1$ создается снеговик номер i . Изначально имеется пустой снеговик с номером ноль.

Все числа во входном файле целые.

Формат выходных данных

Выведите суммарную массу построенных снеговиков.

Пример

стандартный ввод	стандартный вывод
8	74
0 1	
1 5	
2 4	
3 2	
4 3	
5 0	
6 6	
1 0	

Задача Н. Персистентная очередь

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Реализуйте персистентную очередь.

Формат входных данных

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

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

Пример

стандартный ввод	стандартный вывод
10	1
1 0 1	2
1 1 2	3
1 2 3	1
1 2 4	2
-1 3	4
-1 5	
-1 6	
-1 4	
-1 8	
-1 9	