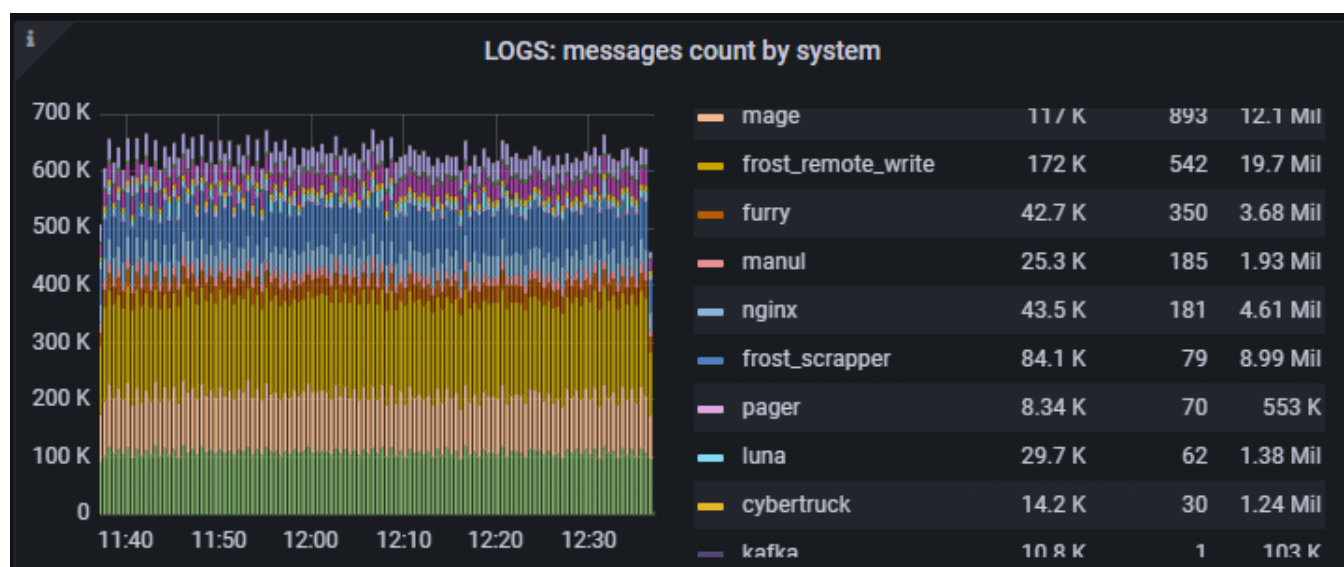


Задача A. Sage

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

Побывав в команде разработки поиска по коду, Петя понял, что искать код – это не его. Но это не беда, ведь в Т-Банке много других команд. Ему приглянулся сервис Sage^[1] – платформа мониторинга, куда все сервисы Т-Банка отправляют свои логи (текст, который вы печатаете при помощи `cout/print`), данные для графиков и прочее.

Пете разумеется доверили самую ответственную задачу: научиться считать количество логов на разных временных промежутках, чтобы потом строить красивые графики, типа того, что ниже.



Пример графика с логами

Петя тут же попытался формализовать задачу. Пусть в момент времени t есть a_t логов (время считаем дискретным). Пусть пользователь пытается построить график для промежутка $[l, r]$. Тогда нужно найти $\sum_{t=l}^r a_t$. Петя разумеется, знает что такая задача решается с помощью префиксных сумм. Но, как оказалось, есть нюансы. Из-за сбоев в сети (или просто тормозов интернета) логи могут приходить с большим опозданием. Более того, некоторые логи могут дублироваться (если интересно почему, почитайте про брокеры сообщений и гарантии *exactly once/at least once* и прочее). Но, благо, есть отдельная система, которая фильтрует дубликаты (правда тоже не сразу, а через какое-то время). Короче говоря, количество логов в разные моменты времени может меняться и это тоже нужно учитывать.

Формализовать задачу у Пети получилось хорошо, а вот придумать решение с чем-то кроме префиксных сумм не получается. Петя просит помощи у вас в решении задачи!

[1] <https://www.tbank.ru/career/technologies/sage/>

Формат входных данных

Первая строка содержит два числа n и m ($1 \leq n, m \leq 10^5$) – количество «моментов времени» и число запросов.

Следующая строка содержит n чисел a_i – количество логов в i -м моменте времени ($0 \leq a_i \leq 10^9$). Далее следует описание операций. Описание каждой операции имеет следующий вид:

- 1 i v – поменять количество логов в i -м фрагменте времени на v ($0 \leq i < n, 0 \leq v \leq 10^9$).
- 2 l r – посчитать сумму по количеству логов на отрезке от l до $r - 1$ ($0 \leq l < r \leq n$).

Формат выходных данных

Для каждой операции второго типа выведите соответствующее количество логов на отрезке.

Пример

стандартный ввод	стандартный вывод
5 5	11
5 4 2 3 5	8
2 0 3	14
1 1 1	
2 0 3	
1 3 1	
2 0 5	

Задача В. Число минимумов на отрезке

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

Теперь измените код дерева отрезков, чтобы кроме минимума на отрезке считалось также и число элементов, равных минимуму.

Формат входных данных

Первая строка содержит два числа n и m ($1 \leq n, m \leq 10^5$) — размер массива и число операций. Следующая строка содержит n чисел a_i — начальное состояние массива ($0 \leq a_i \leq 10^9$). Далее следует описание операций. Описание каждой операции имеет следующий вид:

- 1 i v — присвоить элементу с индексом i значение v ($0 \leq i < n, 0 \leq v \leq 10^9$).
- 2 l r — найти минимум и число элементов, равных минимуму, среди элементов с индексами от l до $r - 1$ ($0 \leq l < r \leq n$).

Формат выходных данных

Для каждой операции второго типа выведите два числа — минимум на заданном отрезке и число элементов, равных этому минимуму.

Пример

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

Задача С. K -я единица

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

В этой задаче вам нужно добавить в дерево отрезков операцию нахождения k -й единицы.

Формат входных данных

Первая строка содержит два числа n и m ($1 \leq n, m \leq 10^5$) – размер массива и число операций. Следующая строка содержит n чисел a_i – начальное состояние массива ($a_i \in \{0, 1\}$). Далее следует описание операций. Описание каждой операции имеет следующий вид:

- 1 i – изменить элемент с индексом i на противоположный.
- 2 k – найти k -ю единицу (единицы нумеруются с 0, гарантируется, что в массиве достаточное количество единиц).

Формат выходных данных

Для каждой операции второго типа выведите индекс соответствующей единицы (все индексы в этой задаче от 0).

Пример

стандартный ввод	стандартный вывод
5 7	0
1 1 0 1 0	1
2 0	3
2 1	3
2 2	1
1 2	
2 3	
1 0	
2 0	

Задача D. Первый элемент не меньше X

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

В этой задаче вам нужно добавить в дерево отрезков операцию нахождения по данным x и l минимального индекса j , для которого $j \geq l$ и $a[j] \geq x$.

Формат входных данных

Первая строка содержит два числа n и m ($1 \leq n, m \leq 10^5$) – размер массива и число операций. Следующая строка содержит n чисел a_i – начальное состояние массива ($0 \leq a_i \leq 10^9$). Далее следует описание операций. Описание каждой операции имеет следующий вид:

- 1 i v – изменить элемент с индексом i на v ($0 \leq i < n, 0 \leq v \leq 10^9$).
- 2 x l – найти минимальный индекс j , для $j \geq l$ и $a[j] \geq x$ ($0 \leq x \leq 10^9, 0 \leq l < n$). Если такого элемента нет, выведите -1 . Индексы начинаются с 0.

Формат выходных данных

Для каждой операции второго типа выведите ответ на запрос.

Пример

стандартный ввод	стандартный вывод
5 7	1
1 3 2 4 3	3
2 3 0	2
2 3 2	-1
1 2 5	3
2 4 1	
2 5 4	
1 3 7	
2 6 1	

Задача Е. Число возрастающих подпоследовательностей

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

Задана последовательность из n чисел. Необходимо найти число возрастающих подпоследовательностей наибольшей длины заданной последовательности a_1, \dots, a_n . Так как это число может быть достаточно большим, необходимо найти остаток от его деления на $10^9 + 7$.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$). Вторая строка входного файла содержит n целых чисел: a_1, a_2, \dots, a_n . Все a_i не превосходят 10^9 по абсолютной величине.

Формат выходных данных

В выходной файл выведите ответ на задачу.

Примеры

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

Задача F. Противник слаб.

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Римляне снова наступают. На этот раз их гораздо больше, чем персов, но Шапур готов победить их. Он говорит: «Лев никогда не испугается сотни овец».

Не смотря на это, Шапур должен найти слабость римской армии, чтобы победить ее. Как вы помните, Шапур – математик, поэтому он определяет, насколько слаба армия, как число – степень слабости.

Шапур считает, что степень слабости армии равна количеству таких троек i, j, k , что $i < j < k$ и $a_i > a_j > a_k$, где a_x – сила человека, стоящего в строю на месте с номером x .

Помогите Шапуру узнать, насколько слаба армия римлян.

Формат входных данных

В первой строке записано одно целое число n ($3 \leq n \leq 10^5$) – количество солдат в римской армии. Следующая строка содержит n целых чисел a_i ($1 \leq i \leq n, 1 \leq a_i \leq 10^9$) – силы людей в римской армии.

Формат выходных данных

Выведите одно число – степень слабости римской армии.

Примеры

стандартный ввод	стандартный вывод
3 3 2 1	1
3 2 3 1	0
4 10 8 3 1	4
4 1 5 4 3	1