

Учебно-тренировочные сборы к РОИ-2017, группа А2. $\sqrt{\text{Тур}}$
14 марта 2018 года

Тур проводится по АСМ-системе, все задачи оцениваются только целиком. Информация о подзадачах оставлена в качестве подсказки к структуре тестов.

В некоторых задачах стоит достаточно щадящие ограничения по времени. Пожалуйста, пишите решения с *правильными* асимптотиками.

Не забудьте прочитать *все* задачи.

Содержание

Задача А. Сумма	2
Задача В. И снова сумма...	3
Задача С. Инверсии отрезка	4
Задача D. Фаброзавры-дизайнеры	5
Задача Е. Substring Query	6
Задача F. На далёкой Амазонке	7
Задача G. Жесть	9
Задача H. Машинное обучение	10

Задача А. Сумма

Имя входного файла: `sum.in`
Имя выходного файла: `sum.out`
Ограничение по времени: 0.5 секунды
Ограничение по памяти: 256 мегабайт

Разбейте массив на кусочки фиксированного размера.

Дан массив из N элементов, нужно научиться находить сумму чисел на отрезке.

Формат входных данных

Первая строка входного файла содержит два целых числа N и K — количество чисел в массиве и количество запросов ($1 \leq N \leq 100\,000$, $0 \leq K \leq 100\,000$). Следующие K строк содержат запросы.

1. `A i x` — присвоить i -му элементу массива значение x ($1 \leq i \leq n$, $0 \leq x \leq 10^9$).
2. `Q l r` — найти сумму чисел в массиве на позициях от l до r ($1 \leq l \leq r \leq n$).

Изначально в массиве живут нули.

Формат выходных данных

На каждый запрос вида `Q l r` нужно вывести единственное число — сумму на отрезке.

Примеры

sum.in	sum.out
5 9	0
A 2 2	2
A 3 1	1
A 4 2	2
Q 1 1	0
Q 2 2	5
Q 3 3	
Q 4 4	
Q 5 5	
Q 1 5	

Задача В. И снова сумма...

Имя входного файла: `sum2.in`
Имя выходного файла: `sum2.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Используйте метод отложенных запросов.

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с которым разрешается производить следующие операции:

- $\text{add}(i)$ — добавить в множество S число i (если он там уже есть, то множество не меняется);
- $\text{sum}(l, r)$ — вывести сумму всех элементов x из S , которые удовлетворяют неравенству $l \leq x \leq r$.

Формат входных данных

Исходно множество S пусто. Первая строка входного файла содержит n — количество операций ($1 \leq n \leq 300\,000$). Следующие n строк содержат операции. Каждая операция имеет вид либо «+ i », либо «? l r ». Операция «? l r » задает запрос $\text{sum}(l, r)$.

Если операция «+ i » идет во входном файле в начале или после другой операции «+», то она задает операцию $\text{add}(i)$. Если же она идет после запроса «?», и результат этого запроса был y , то выполняется операция $\text{add}((i + y) \bmod 10^9)$.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до 10^9 .

Формат выходных данных

Для каждого запроса выведите одно число — ответ на запрос.

Пример

sum2.in	sum2.out
6	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	

Задача С. Инверсии отрезка

Имя входного файла: `invseg.in`
Имя выходного файла: `invseg.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Дана перестановка p из n чисел. Ваша задача — быстро обрабатывать запросы вида *количество инверсий на отрезке данной перестановки*. Напомним, инверсией называется пара индексов (i, j) : $i < j$, $p_i > p_j$.

Формат входных данных

На первой строке целое число n .

На второй строке сама перестановка.

На третьей строке количество запросов q .

Следующие q строк содержат пары целых чисел l_i, r_i ($1 \leq l_i \leq r_i \leq n$).

Формат выходных данных

Для каждого из q запросов выведите ответ на отдельной строке.

Система оценки

Подзадача 1 (50 баллов) $1 \leq n, q \leq 10^5$ $l_i \leq l_{i+1}, r_i \leq r_{i+1}$.

Подзадача 2 (50 баллов) $1 \leq n, q \leq 20\,000$ l_i и r_i произвольны

Пример

invseg.in	invseg.out
5	3
5 4 3 1 2	9
3	2
1 3	
1 5	
3 5	

Задача D. Фаброзавры-дизайнеры

Имя входного файла: `fabro.in`
Имя выходного файла: `fabro.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Фаброзавры известны своим тонким художественным вкусом и увлечением ландшафтным дизайном. Они живут около очень живописной реки и то и дело перестраивают тропинку, идущую вдоль реки: либо насыпают дополнительной земли, либо срывают то, что есть. Для того, чтобы упростить эти работы, они поделили всю тропинку на горизонтальные участки, пронумерованные от 1 до N , и их переделки устроены всегда одинаково: они выбирают часть дороги от L -ого до R -ого участка (включительно) и изменяют (увеличивают или уменьшают) высоту на всех этих участках на одну и ту же величину (если до начала переделки высоты были разными, то и после переделки они останутся разными).

Поскольку, как уже говорилось, у фаброзавров тонкий художественный вкус, каждый из них считает, что их река лучше всего выглядит с определенной высоты. Поэтому им хочется знать, есть ли поблизости от их дома место на тропинке, где высота на их взгляд оптимальна. Помогите им в этом разобраться.

Формат входных данных

Первая строка входного файла содержит два числа N и M — длину дороги и количество запросов соответственно ($1 \leq N, M \leq 10^5$). На второй строке содержатся N чисел, разделенных пробелами — начальные высоты соответствующих частей дороги; высоты не превосходят 10^4 по модулю. В следующих M строках содержатся запросы по одному на строке.

Запрос $+ L R X$ означает, что высоту частей дороги от L -ой до R -ой (включительно) нужно изменить на X . При этом $1 \leq L \leq R \leq N$, а $|X| \leq 10^4$.

Запрос $? L R X$ означает, что нужно проверить, есть ли между L -ым и R -ым участками (включая эти участки) участок, где дорога проходит точно на высоте X . Гарантируется, что $1 \leq L \leq R \leq N$, а $|X| \leq 10^9$.

Формат выходных данных

На каждый запрос второго типа нужно вывести в выходной файл на отдельной строке одно слово «YES» (без кавычек), если нужный участок существует, и «NO» в противном случае.

Примеры

fabro.in	fabro.out
10 5	NO
0 1 1 3 3 3 2 0 0 1	YES
? 3 5 2	YES
+ 1 4 1	
? 3 5 2	
+ 7 10 2	
? 9 10 3	

Задача Е. Substring Query

Имя входного файла: `str-qry.in`
Имя выходного файла: `str-qry.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

У Бобо есть n строк S_1, S_2, \dots, S_n . Однажды его друг Ййи пришёл и попросил ответить на q вопросов: сколько строк среди $S_{l_i}, S_{l_i+1}, \dots, S_{r_i}$ содержат P_i как подстроку?

Помогите Бобо получить ответ.

Формат входных данных

На первой строке 2 целых числа n, q ($1 \leq n, q \leq 200\,000$). Каждая из следующих n строк содержит ровно одну строку S_i ($|S_1| + |S_2| + \dots + |S_n| \leq 200\,000$). Последние q строк содержат запросы. Каждый запрос задаётся двумя целыми числами l_i, r_i и строкой P_i . ($1 \leq l_i \leq r_i \leq n, |P_1| + |P_2| + \dots + |P_n| \leq 200\,000$) Все строки состоят из букв “a” и “b”.

Формат выходных данных

На каждый вопрос выведите одно число – ответ на вопрос.

Примеры

str-qry.in	str-qry.out
4 2	2
a	2
b	
ab	
bab	
1 3 a	
1 4 ab	

Задача F. На далёкой Амазонке

Имя входного файла: `treeeg.in`
Имя выходного файла: `treeeg.out`
Ограничение по времени: 10 секунд
Ограничение по памяти: 256 мегабайт

В бассейне далёкой реки Амазонки расположены N городов, пронумерованных для удобства целыми числами от 1 до N . Всем известно, что местные леса непроходимы, и передвижение возможно только по рекам. Как следствие, схема соединения городов является деревом.

К несчастью, в этом году в бассейне далёкой Амазонки не на шутку разошлась эпидемия новой болезни — крабового гриппа. То и дело поступает информация о новых заболевших. Поначалу справляться с ней было легко, но вскоре почти все больницы были переполнены, и сейчас пациентов может принимать только госпиталь, находящийся в городе 1.

Для удобства граждан была открыта горячая линия, куда первым делом необходимо обратиться при появлении симптомов крабового (его ещё часто называют раковым) гриппа. Вам необходимо написать программу, которая будет отвечать на обращения пострадавших, учитывая при этом информацию о работающих больницах. Вам ещё повезло, что вы знаете все запросы заранее!

Более формально, поступают запросы трёх видов:

- «+ v » — госпиталь города v снова может принимать больных. Гарантируется, что в момент перед этим запросом госпиталь города v не работал.
- «- v » — госпиталь города v не может больше принимать больных. Гарантируется, что в момент перед этим запросом госпиталь города v работал.
- «? v » — заболел человек в городе v , необходимо сообщить ему расстояние до ближайшего города с работающим госпиталем (в идеале неплохо бы ещё и сказать номер этого города, но этим пусть занимаются ваши коллеги). Гарантируется, что в момент такого запроса имеется хотя бы один работающий госпиталь.

Формат входных данных

В первой строке находится единственное число N — количество городов ($1 \leq N \leq 300\,000$). Следующие $N - 1$ строк содержат информацию о соединениях между городами в формате « $u \ v \ l$ », что означает соединение между городами u и v длиной l километров ($1 \leq u, v \leq N$, $1 \leq l \leq 1000$). Направлением течения можно пренебречь и считать, что время движения зависит только от расстояний.

Далее на отдельной строке записано число Q — количество запросов. Следующие Q строк содержат описание запросов в формате « $s \ v$ », где s — это один из трёх символов «+», «-» и «?», а v — номер города ($1 \leq v \leq N$).

Формат выходных данных

Для каждого запроса вида «? v » выведите на отдельной строке одно число — расстояние в километрах до ближайшего города с работающим госпиталем.

Пример

treeeg.in	treeeg.out
5	6
1 2 2	4
2 3 3	7
3 4 1	
3 5 4	
5	
? 4	
+ 5	
? 3	
- 1	
? 2	

Подзадача 1 (баллы: 20)

Ограничения: $1 \leq N, Q \leq 3000$.

Подзадача 2 (баллы: 40)

Ограничения: $1 \leq N, Q \leq 100\,000$.

Задача G. Жесть

Имя входного файла: `sqrtrev.in`
Имя выходного файла: `sqrtrev.out`
Ограничение по времени: 10 секунд
Ограничение по памяти: 256 мегабайт

Дан массив из N чисел. Нужно уметь обрабатывать 3 типа запросов:

- `get(L, R, x)` — сказать, сколько элементов отрезка массива $[L..R]$ не меньше x .
- `set(L, R, x)` — присвоить всем элементам массива на отрезке $[L..R]$ значение x .
- `reverse(L, R)` — перевернуть отрезок массива $[L..R]$.

Формат входных данных

Число N ($1 \leq N \leq 10^5$) и массив из N чисел. Далее число запросов M ($1 \leq M \leq 10^5$) и M запросов. Формат описания запросов предлагается понять из примера. Для всех отрезков верно $1 \leq L \leq R \leq N$. Исходные числа в массиве и числа x в запросах — целые от 0 до 10^9 .

Формат выходных данных

Для каждого запроса типа `get` нужно вывести ответ.

Пример

<code>sqrtrev.in</code>	<code>sqrtrev.out</code>
5	3
1 2 3 4 5	1
6	3
get 1 5 3	1
set 2 4 2	
get 1 5 3	
reverse 1 2	
get 2 5 2	
get 1 1 2	

Задача Н. Машинное обучение

Имя входного файла: `cnt-mex.in`
Имя выходного файла: `cnt-mex.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

На курсе машинного обучения вам выдали первое домашнее задание — вам предстоит проанализировать некоторый массив из n чисел.

В частности, вы интересуетесь так называемой *равномерностью* массива. Предположим, что в массиве число b_1 встречается k_1 раз, b_2 — k_2 раз, и т.д. Тогда *равномерностью* массива называется такое минимальное целое число $c \geq 1$, что $c \neq k_i$ для любого i .

В рамках вашего исследования вы хотите последовательно проделать q операций.

- Операция $t_i = 1$, l_i , r_i задаёт запрос исследования. Необходимо вывести равномерность массива, состоящего из элементов на позициях от l_i до r_i включительно.
- Операция $t_i = 2$, p_i , x_i задаёт запрос уточнения данных. Начиная с этого момента времени p_i -му элементу массива присваивается значения x_i .

Формат входных данных

Первая строка содержит n и q ($1 \leq n, q \leq 100\,000$) — размер массива и число запросов соответственно.

Во второй строке записаны ровно n чисел — a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Каждая из оставшихся q строк задаёт очередной запрос.

Запрос первого типа задаётся тремя числами $t_i = 1$, l_i , r_i , где $1 \leq l_i \leq r_i \leq n$ — границы соответствующего отрезка.

Запрос второго типа задаётся тремя числами $t_i = 2$, p_i , x_i , где $1 \leq p_i \leq n$ — позиция в которой нужно заменить число, а $1 \leq x_i \leq 10^9$ — его новое значение

Формат выходных данных

Для каждого запроса первого типа выведите одно число — равномерность соответствующего отрезка массива.

Пример

<code>cnt-mex.in</code>	<code>cnt-mex.out</code>
10 4	2
1 2 3 1 1 2 2 2 9 9	3
1 1 1	2
1 2 8	
2 7 1	
1 2 8	

Замечание

Первый запрос состоит из ровно одного элемента — 1. Минимальное подходящее $c = 2$.

Отрезок второго запроса состоит из четырёх 2, одной 3 и двух 1. Минимальное подходящее $c = 3$.

Отрезок четвёртого запроса состоит из трёх 1, трёх 2 и одной 3. Минимальное подходящее $c = 2$.