

Содержание

Must have	2
Задача 23А. Фаброзавры-дизайнеры [1.2, 256]	2
Задачи здорового человека	3
Задача 23В. Подстроки со сдвигом [0.6, 256]	3
Задача 23С. Substring Query [0.6, 256]	4
Задача 23D. Count Online [2.0, 256]	5
Задача 23Е. Переворачивания [1.5, 256]	6
Для искателей острых ощущений	7
Задача 23F. Connect and Disconnect [1.5, 256]	7
Задача 23G. Опять k-я статистика [2.5, 256]	8
Задача 23H. Прямоугольные запросы [3.5, 256]	9

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же `stdin`), вывести ответ нужно в **стандартный поток вывода** (он же `stdout`).

Обратите внимание на GNU C++ компиляторы с суффиксом `inc`.

Подни можно пользоваться **дополнительной библиотекой** (`optimization.h`).

То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.

И быструю аллокацию памяти (ускоряет `vector-set-map-весь-STL`): **пример**.

Для тех, кто хочет разобраться, как всё это работает.

Короткая версия быстрого ввода-вывода (**тык**) и короткая версия аллокатора (**тык**).

Must have

Задача 23А. Фаброзавры-дизайнеры [1.2, 256]

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

stdin	stdout
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	

Подсказка по решению

Базовая задача с лекции.

Задачи здорового человека

Задача 23В. Подстроки со сдвигом [0.6, 256]

Вам даны K текстов. Все тексты имеют одинаковую длину.

Ваша задача — научиться искать подстроку со сдвигом. Подстрока S со сдвигом a_1, a_2, \dots, a_K входит в набор из K текстов T_1, T_2, \dots, T_K , если существует такое число x , что для всех i $LCP(T_i + a_i + x, S) \geq |S|$. Где LCP — длина наибольшего общего префикса, $(T_i + j)$ — j -й суффикс строки T_i , $|S|$ — длина строки S .

Формат входных данных

Число K от 1 до 10 и K текстов (длины текстов одинаковы и лежат от 1 до 10^5). Далее M от 1 до 10^5 — число запросов и сами запросы. Каждый запрос это строка и K чисел от -10^9 до 10^9 . Суммарная длина всех строк в запросах не более 10^5 . Все строки и тексты состоят только из маленьких символов английского алфавита. **Все строки S по всем запросам различны.**

Формат выходных данных

Для каждого запроса выведите NO или YES x (x — величина из условия).

Пример

stdin	stdout
3	YES 0
abacabaa	YES 5
ababbbbaa	NO
aababbbb	YES 0
4	
a 0 0 1	
b 0 0 0	
ba 0 0 -1	
aa 6 6 0	

Подсказка по решению

Это анекдот на тему «различных длин строк мало» \Rightarrow можно перебрать длину.

Хеши и хеш-таблицы скрасят вашу жизнь в минуты отчаяния.

Задача 23С. Substring Query [0.6, 256]

У Бобо есть n строк S_1, S_2, \dots, S_n . Однажды его друг Ййии 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

stdin	stdout
4 2	2
a	2
b	
ab	
bab	
1 3 a	
1 4 ab	

Подсказка по решению

Это анекдот на тему «различных длин строк мало» \Rightarrow можно перебрать длину.
Хеши и хеш-таблицы скрасят вашу жизнь в минуты отчаяния.

Задача 23D. Count Online [2.0, 256]

Вам дано множество точек на плоскости.

Нужно уметь отвечать на два типа запросов:

○ ? $x_1 y_1 x_2 y_2$ — сказать, сколько точек лежит в прямоугольнике $[x_1..x_2] \times [y_1..y_2]$. Точки на границе и в углах тоже считаются. $x_1 \leq x_2, y_1 \leq y_2$.

○ + $x y$ — добавить в множество точку $(x + \text{res} \% 100, y + \text{res} \% 101)$. Где res — ответ на последний запрос вида ?, а $\%$ — операция взятия по модулю.

Формат входных данных

Число точек N ($1 \leq N \leq 50\,000$). Далее N точек.

Число запросов Q ($1 \leq Q \leq 100\,000$). Далее Q запросов. Все координаты от 0 до 10^9 .

Формат выходных данных

Для каждого запроса GET одно целое число — количество точек внутри прямоугольника.

Пример

stdin	stdout
5	3
0 0	3
1 0	1
0 1	0
1 1	0
1 1	3
9	
? 0 1 1 2	
+ 1 2	
+ 2 2	
? 1 0 2 2	
? 0 0 0 0	
+ 3 3	
? 3 3 3 3	
? 4 3 4 3	
? 4 4 5 5	

Замечание

На самом деле добавлялись точки $(4, 5), (5, 5), (4, 4)$.

Подсказка по решению

Корневая по запросам. А внутри то, что вы хорошо умеете.

Задача 23Е. Переворачивания [1.5, 256]

Учитель физкультуры школы с углубленным изучением предметов уже давно научился считать суммарный рост всех учеников, находящихся в ряду на позициях от l до r . Но дети играют с ним злую шутку. В некоторый момент дети на позициях с l по r меняются местами. Учитель заметил, что у детей не очень богатая фантазия, поэтому они всегда «переворачивают» этот отрезок, т. е. l меняется с r , $l + 1$ меняется с $r - 1$ и так далее. Но учитель решил не ругать детей за их хулиганство, а все равно посчитать суммарный рост на всех запланированных отрезках.

Формат входных данных

В первой строке записано два числа n и m ($1 \leq n, m \leq 200\,000$) — количество детей в ряду и количество событий, произошедших за все время. Во второй строке задано n натуральных чисел — рост каждого школьника в порядке следования в ряду. Рост детей не превосходит $2 \cdot 10^5$. Далее в m строках задано описание событий: три числа q, l, r в каждой строке ($0 \leq q \leq 1, 1 \leq l \leq r \leq n$). Число q показывает тип события: 0 показывает необходимость посчитать и вывести суммарный рост школьников на отрезке $[l, r]$; 1 показывает то, что дети на отрезке $[l, r]$ «перевернули» свой отрезок. Все числа во входном файле целые.

Формат выходных данных

Для каждого события типа 0 выведите единственное число на отдельной строке — ответ на этот запрос.

Пример

stdin	stdout
5 6	15
1 2 3 4 5	9
0 1 5	8
0 2 4	7
1 2 4	10
0 1 3	
0 4 5	
0 3 5	

Подсказка по решению

Корневая по массиву. Нужно split-ить отрезки и иногда делать rebuild.

Для искателей острых ощущений

Задача 23F. Connect and Disconnect [1.5, 256]

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.

Граф является неориентированным. Изначально граф пустой.

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 ‘?’ выведите количество компонент связности в момент запроса.

Пример

stdin	stdout
5 11	5
?	1
+ 1 2	1
+ 2 3	2
+ 3 4	
+ 4 5	
+ 5 1	
?	
- 2 3	
?	
- 4 5	
?	

Подсказка по решению

Корневая по запросам. Было на лекции, похожее было в практике.

Задача 23G. Опять k-я статистика [2.5, 256]

Изначально вам дан массив целых чисел.

Нужно уметь отвечать на три запроса:

- $+ i x$ — Вставить на i -ю позицию число x (размер массива увеличивается на 1)
- $- i$ — Удалить число на i -й позиции (размер массива уменьшается на 1)
- $? L R x$ — Сказать, сколько чисел y на позициях $L \leq i \leq R$ таких, что $y \leq x$ ($|x| \leq 10^9$)

Все индексы i , L , R нумеруются с нуля. Все числа в запросах целые. Все запросы корректны. Пример запроса: “+ 0 x” означает “добавление x в начало массива”. Исходное число элементов в массиве $-0 \leq N \leq 10^5$, числа в массиве по модулю не превышают 10^9 . Число запросов $-1 \leq K \leq 10^5$.

Пример

stdin	stdout
10	1
455184306 359222813 948543704	2
914773487 861885581 253523	2
770029097 193773919 581789266	0
457415808	2
- 1	
? 2 5 527021001	
? 0 5 490779085	
? 0 5 722862778	
+ 9 448694272	
- 5	
? 1 2 285404014	
- 4	
? 3 4 993634734	
+ 0 414639071	

Подсказка по решению

Корневая по массиву. Нужно split-ить отрезки и иногда делать rebuild.

Задача 23Н. Прямоугольные запросы [3.5, 256]

Даны N точек на плоскости, у каждой точки есть ценность. Нужно быстро обрабатывать запросы двух типов:

- Присвоить всем точкам в области $[x_1..x_2] \times [y_1..y_2]$ ценность K .
- Найти точку с минимальной ценностью в области $[x_1..x_2] \times [y_1..y_2]$.

Формат входных данных

Вам даны число точек N ($1 \leq N \leq 262144$) и N точек.

Каждая точка задается тремя числами — x, y , начальная ценность.

Далее следует число запросов M ($1 \leq M \leq 10^4$) и M запросов в формате

“= $x_1 y_1 x_2 y_2 value$ ” для присваивания и “? $x_1 y_1 x_2 y_2$ ” для взятия минимума.

Все координаты от -10^9 до 10^9 . Все ценности от 0 до 10^9 .

Формат выходных данных

На каждый запрос ? выведите минимальную ценность точек в прямоугольнике.

Если в прямоугольнике нет ни одной точки, выведите NO.

Пример

stdin	stdout
4	2
1 1 1	1
-1 1 1	NO
-1 -1 1	0
1 -1 1	
7	
= 0 0 3 3 2	
= -3 -3 0 0 2	
? 0 0 3 3	
? -3 -3 3 3	
= -1 -1 1 1 0	
? 0 0 0 0	
? -1000 -1000 1000 1000	

Подсказка по решению

Корневая по массиву. Нужно split-ить отрезки и иногда делать rebuild.