

Задача А. RMQ

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

Реализуйте структуру данных, которая на данном массиве из N целых чисел позволяет узнать максимальное значение на этом массиве и индекс элемента, на котором достигается это максимальное значение.

Формат входных данных

В первой строке вводится натуральное число N ($1 \leq N \leq 10^5$) – количество элементов в массиве. В следующей строке содержатся N целых чисел, не превосходящих по модулю 10^9 – элементы массива. Гарантируется, что в массиве нет одинаковых элементов. Далее идет число K ($0 \leq K \leq 10^5$) – количество запросов к структуре данных. Каждая из следующих K строк содержит два целых числа l и r ($1 \leq l \leq r \leq N$) – левую и правую границы отрезка в массиве для данного запроса.

Формат выходных данных

Для каждого из запросов выведите два числа: наибольшее значение среди элементов массива на отрезке от l до r и индекс одного из элементов массива, принадлежащий отрезку от l до r , на котором достигается этот максимум.

Пример

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

Задача В. Нолики

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

Дедус любит давать своим ученикам сложные задачки. На этот раз он придумал такую задачу: Рейтинг всех его учеников записан в массив A . Запросы Дедуса таковы:

1. Изменить рейтинг i -го ученика на число x
2. Найти максимальную последовательность подряд идущих ноликов в массиве A на отрезке $[l, r]$.

Помогите бедным фиксикам избежать зверского наказания за нерешение задачи на этот раз.

Формат входных данных

В первой строке входного файла записано число N ($1 \leq N \leq 500\,000$) – количество учеников. Во второй строке записано N чисел – их рейтинги, числа по модулю не превосходящие 1000 (по количеству задач, которые ученик решил или не решил за время обучения). В третьей строке записано число M ($1 \leq M \leq 50\,000$) – количество запросов. Каждая из следующих M строк содержит описание запросов:

«UPDATE i x » – обновить i -ый элемент массива значением x ($1 \leq i \leq N$, $|x| \leq 1000$)

«QUERY l r » – найти длину максимальной последовательности из нулей на отрезке с l по r . ($1 \leq l \leq r \leq N$)

Формат выходных данных

В выходной файл выведите ответы на запросы «QUERY» в том же порядке, что и во входном файле

Пример

стандартный ввод	стандартный вывод
5	2
328 0 0 0 0	1
5	1
QUERY 1 3	
UPDATE 2 832	
QUERY 3 3	
QUERY 2 3	
UPDATE 2 0	

Задача С. Поиск максимума

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Реализуйте структуру данных для эффективного вычисления номера максимального из нескольких подряд идущих элементов массива.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100\,000$) — количество чисел в массиве.

Во второй строке вводятся N чисел от 1 до 100 000 — элементы массива.

В третьей строке вводится одно натуральное число K ($1 \leq K \leq 3\,000\,000$) — количество запросов на вычисление максимума.

В следующих K строках вводится по два числа — номера левого и правого элементов отрезка массива (считается, что элементы массива нумеруются с единицы).

Формат выходных данных

Для каждого запроса выведите индекс максимального элемента на указанном отрезке массива. Если максимальных элементов несколько, выведите любой их них.

Числа выводите в одну строку через пробел.

Пример

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

Задача D. НОД на подотрезках с изменением элемента

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

Реализуйте эффективную структуру данных, позволяющую изменять элементы массивы и вычислять НОД нескольких подряд идущих элементов.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100000$) – количество чисел в массиве.

Во второй строке вводятся N чисел от 0 до 100000 – элементы массива.

В третьей строке вводится одно натуральное число M ($1 \leq M \leq 30000$) – количество запросов.

Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (s – вычислить НОД, u – обновить значение элемента).

Следом за s вводятся два числа – номера левой и правой границы отрезка.

Следом за u вводятся два числа – номер элемента и его новое значение.

Формат выходных данных

Для каждого запроса s выведите результат. Все числа выводите в одну строку через пробел.

Пример

стандартный ввод	стандартный вывод
5	2 4 4 32
2 8 4 16 12	
5	
s 1 5	
s 4 5	
u 3 32	
s 2 5	
s 3 3	

Задача Е. Марио и трубы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 256 мегабайт

Марио собирается проходить уровень, состоящий из N последовательно расположенных труб, высота i -й трубы — a_i . Он еще не знает, где он будет располагаться изначально, и куда ему надо добраться, поэтому хочет рассмотреть несколько вариантов.

Находясь на трубе, Марио может переместиться только на соседние трубы слева и справа (если они существуют). Спускаться он может с любой высоты, также он может перемещаться между одинаковыми трубами. Подниматься Марио может только на трубу, высота которой больше высоты текущей на 1. Более формально, Марио может переместиться с трубы i на трубу j , если $|i - j| = 1$ и $a_j - a_i \leq 1$.

Однако злой динозавр Боузер хочет помешать Марио пройти уровень, для чего иногда увеличивает высоту нескольких подряд идущих труб на одно число k . Теперь Марио не может понять, удастся ли ему пройти уровень и поэтому просит вас обрабатывать два типа запросов — Боузер изменяет высоту некоторых труб, и Марио пытается пройти от одной трубы до другой.

Формат входных данных

В первой строке заданы два целых числа N и M — число труб и число запросов соответственно ($2 \leq N \leq 3 \cdot 10^5, 1 \leq M \leq 10^6$).

Следующая строка содержит N целых чисел a_i — высоты труб на уровне ($1 \leq a_i \leq 10^9$).

Далее идут M строк, содержащие описание запросов. Каждая строка имеет вид:

- $1 \ x \ y$ — может ли Марио пройти от трубы с номером x до трубы с номером y ($1 \leq x, y \leq N$). Гарантируется, что номера x и y не совпадают.
- $2 \ l \ r \ d$ — Боузер увеличивает высоты труб с l -й до r -й на величину d ($1 \leq l \leq r \leq N, -10^9 \leq d \leq 10^9$).

Формат выходных данных

Для каждого запроса первого типа нужно на отдельной строке вывести «YES», если Марио может пройти от одной трубы до другой и «NO» в противном случае (без кавычек).

Система оценки

Подзадача	Баллы	Дополнительные ограничения	Оценка	Необходимые подзадачи
0	0	Тесты из условия	подзадача	—
1	20	$1 \leq N, M \leq 10^4$	подзадача	—
2	60	$1 \leq N \leq 10^5, 1 \leq M \leq 3 \cdot 10^5$	подзадача	1
3	20	—	подзадача	1, 2

Пример

стандартный ввод	стандартный вывод
5 7	YES
1 2 3 4 5	NO
1 5 1	NO
2 2 4 3	YES
1 5 4	NO
1 1 3	
2 2 3 3	
1 2 4	
1 1 3	

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 армии 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 ($3 \leq n \leq 10^6$) — 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

Задача G. Количество инверсий

Имя входного файла: `inverse.in`
Имя выходного файла: `inverse.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Напишите программу, которая для заданного массива $A = \langle a_1, a_2, \dots, a_n \rangle$ находит количество пар (i, j) таких, что $i < j$ и $a_i > a_j$. Обратите внимание на то, что ответ может не влезать в `int`.

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 100\,000$) — количество элементов массива. Вторая строка содержит n попарно различных элементов массива A — целых неотрицательных чисел, не превосходящих 10^9 .

Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

Примеры

<code>inverse.in</code>	<code>inverse.out</code>
5 6 11 18 28 31	0
8 999994 999989 999982 999972 999969 999961 999954 999950	28

Задача Н. Знакопереживание

Имя входного файла: `signchange.in`
Имя выходного файла: `signchange.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Реализуйте структуру данных из n элементов a_1, a_2, \dots, a_n , поддерживающую следующие операции:

- присвоить элементу a_i значение j ;
- найти знакопереживающую сумму на отрезке от l до r включительно, т. е. $(a_l - a_{l+1} + a_{l+2} - \dots - a_r)$.

Формат входных данных

В первой строке входного файла содержится натуральное число n ($1 \leq n \leq 10^5$) — длина массива. Во второй строке записаны начальные значения элементов — неотрицательные целые числа, не превосходящие 10^4 .

В третьей строке находится натуральное число m ($1 \leq m \leq 10^5$) — количество операций. В последующих m строках записаны операции:

- операция первого типа задаётся тремя числами $0 \ i \ j$ ($1 \leq i \leq n, 1 \leq j \leq 10^4$).
- операция второго типа задаётся тремя числами $1 \ l \ r$ ($1 \leq l \leq r \leq n$).

Формат выходных данных

Для каждой операции второго типа выведите на отдельной строке соответствующую знакопереживающую сумму.

Пример

<code>signchange.in</code>	<code>signchange.out</code>
3	-1
1 2 3	2
5	-1
1 1 2	3
1 1 3	
1 2 3	
0 2 1	
1 1 3	

Замечание

TL для Python 4 секунды

Задача I. Без сказок

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 мегабайта

У этой задачи нет легенды. Вам дана последовательность из N целых чисел и M запросов одного из двух типов:

- *change* ps, val — заменить число стоящее на позиции ps числом val .
- *get* l, r — найти подотрезок отрезка $[l, r]$ с максимальной суммой.

Обратите внимание на факт, что по определению, пустой отрезок является подотрезком любого отрезка.

Формат входных данных

Первая строка содержит два целых положительных числа N и M не превосходящих 300 000. Следующая строка содержит N целых чисел - изначальную последовательность. Следующие M строк содержат запросы в формате описанном в условии. Гарантируется, что все запросы корректны и все значения в последовательности в любой момент не превосходят по модулю 10^9 . Используется индексация от 1.

Формат выходных данных

Для каждого запроса *get* выведите одно число — сумму чисел на подотрезке являющемся ответом на данный запрос.

Примеры

<code>stdin</code>	<code>stdout</code>
4 2	2
-5 2 -1 2	3
get 1 2	
get 1 4	