

Задача А. Сумма на отрезке

Имя входного файла: `sum.in`
Имя выходного файла: `sum.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

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 следующие запросы:

- A i x — присвоить i -му элементу массива значение x ($1 \leq i \leq n$, $0 \leq x \leq 10^9$);
- 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

| <code>sum.in</code> | <code>sum.out</code> |
|---------------------|----------------------|
| 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 | |

Замечание

TL для Python 4 секунды

Задача В. RMQ

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

Дан массив $a[1..n]$. Требуется написать программу, обрабатывающую два типа запросов.

- Запрос “`max l r`”. Требуется найти максимум в массиве a от l -ой ячейки до r -ой включительно.
- Запрос “`add l r v`”. Требуется прибавить значение v к каждой ячейке массива a от l -ой до r -ой включительно.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leq n, q \leq 10^5$) – длина массива и число запросов соответственно. Вторая строка содержит n целых чисел a_1, \dots, a_n ($|a_i| \leq 10^5$), задающих соответствующие значения массива. Следующие q строк содержат запросы.

В зависимости от типа запрос может иметь вид либо “`max l r`”, либо “`add l r v`”. При этом $1 \leq l \leq r \leq n$, $|v| \leq 10^5$.

Формат выходных данных

Для каждого запроса вида “`max l r`” требуется в отдельной строке выдать значение соответствующего максимума.

Примеры

| <code>rmq.in</code> | <code>rmq.out</code> |
|---------------------|----------------------|
| 5 3 | 3 |
| 1 2 3 4 -5 | 7 |
| max 1 3 | |
| add 1 2 5 | |
| max 1 3 | |

Задача С. Мега-инверсии

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

Инверсией в перестановке p_1, p_2, \dots, p_N называется пара (i, j) такая, что $i < j$ и $p_i > p_j$. Назовём мега-инверсией в перестановке p_1, p_2, \dots, p_N тройку (i, j, k) такую, что $i < j < k$ и $p_i > p_j > p_k$. Напишите алгоритм для быстрого подсчёта количества мега-инверсий в перестановке.

Формат входных данных

Первая строка входного файла содержит целое число N ($1 \leq N \leq 100\,000$). Следующие N чисел описывают перестановку: p_1, p_2, \dots, p_N ($1 \leq p_i \leq N$), все p_i попарно различны. Числа разделяются переводами строк.

Формат выходных данных

Единственная строка выходного файла должна содержать одно число, равное количеству мега-инверсий в перестановке p_1, p_2, \dots, p_N .

Примеры

| <code>mega.in</code> | <code>mega.out</code> |
|----------------------|-----------------------|
| 4 | 4 |
| 4 | |
| 3 | |
| 2 | |
| 1 | |

Задача D. Ближайшее большее число справа

Имя входного файла: `nearandmore.in`
Имя выходного файла: `nearandmore.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан массив a из n чисел. Нужно обрабатывать запросы:

0. `set(i, x)` – присвоить новое значение элементу массива $a[i] = x$;
1. `get(i, x)` – найти $\min k: k \geq i$ и $a_k \geq x$.

Формат входных данных

Первая строка входных данных содержит два числа: длину массива n и количество запросов m ($1 \leq n, m \leq 200\,000$).

Во второй строке записаны n целых чисел – элементы массива a ($0 \leq a_i \leq 200\,000$).

Следующие m строк содержат запросы, каждый запрос содержит три числа t, i, x . Первое число t равно 0 или 1 – тип запроса. $t = 0$ означает запрос типа `set`, $t = 1$ соответствует запросу типа `get`, $1 \leq i \leq n$, $0 \leq x \leq 200\,000$. Элементы массива нумеруются с единицы.

Формат выходных данных

На каждый запрос типа `get` на отдельной строке выведите соответствующее значение k . Если такого k не существует, выведите -1 .

Примеры

| <code>nearandmore.in</code> | <code>nearandmore.out</code> |
|-----------------------------|------------------------------|
| 4 5 | 1 |
| 1 2 3 4 | 3 |
| 1 1 1 | -1 |
| 1 1 3 | 2 |
| 1 1 5 | |
| 0 2 3 | |
| 1 1 3 | |

Замечание

TL для Python 10 секунд

Задача Е. Атомы

| | |
|-------------------------|--------------|
| Имя входного файла: | atoms.in |
| Имя выходного файла: | atoms.out |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

В лаборатории аномальных материалов антинауочно-исследовательского комплекса «Black Mesa» проводят эксперименты с недавно разработанным графитовым наностержнем. Графитовый наностержень представляет собой n последовательно соединенных атомов углерода, находящихся на одной прямой. Каждый атом имеет определенный заряд.

Для проведения эксперимента, стержень располагают вертикально. Пронумеруем атомы от 1 до n снизу вверх. Между двумя атомами образуется сильная связь, если это соседние атомы и верхний из них имеет заряд ровно на один больше, чем нижний. Иными словами, атомы a и b соединены сильной связью, если $a = b + 1$ и $q_a = q_b + 1$, где q_i — заряд i -го атома. Цепочкой атомов назовем несколько последовательных атомов, соединенных сильными связями.

Вчера был проведен очередной эксперимент. Перед началом эксперимента каждому атому установили определенный заряд: i -му атому установили заряд q_i .

Во время эксперимента ученые проводили действия двух типов:

- у всех атомов с номерами от l_i до r_i , включительно, заряд изменяли на величину d_i ;
- временно разрушали все сильные связи атомов, кроме тех, которые соединяют атомы с номерами от l_i до r_i , включительно, и измеряли длину самой длинной цепочки атомов среди оставшихся сильных связей. Затем восстанавливали все временно разрушенные связи.

Было произведено m действий, однако выяснилось, что в результате побочного эффекта эксперимента запись результатов измерений оказалась утеряна. Для продолжения работы с графитовым наностержнем необходимо восстановить результаты вчерашних измерений. К счастью, сохранился план действий, произведенных во время эксперимента. Помогите ученым продолжить исследования, восстановите результаты измерений.

Формат входных данных

В первой строке находится одно целое число n ($1 \leq n \leq 100\,000$) — количество атомов в наностержне. Во второй строке находятся n чисел q_i ($|q_i| \leq 10^9$) — начальный заряд i -го атома. В третьей строке находится одно целое число m ($0 \leq m \leq 100\,000$) — количество действий в эксперименте. В следующих m строках содержится описание эксперимента.

Если строка начинается с символа «+», очередное действие — изменение заряда атомов. В таком случае, далее в этой строке находятся три целых числа: l_i , r_i и d_i ($1 \leq l_i \leq r_i \leq n$, $|d_i| \leq 10^9$), которые характеризуют это действие.

Если строка начинается с символа «?», очередное действие — второго типа. В таком случае, далее в этой строке находятся два целых числа: l_i и r_i ($1 \leq l_i \leq r_i \leq n$), которые характеризуют это действие.

Формат выходных данных

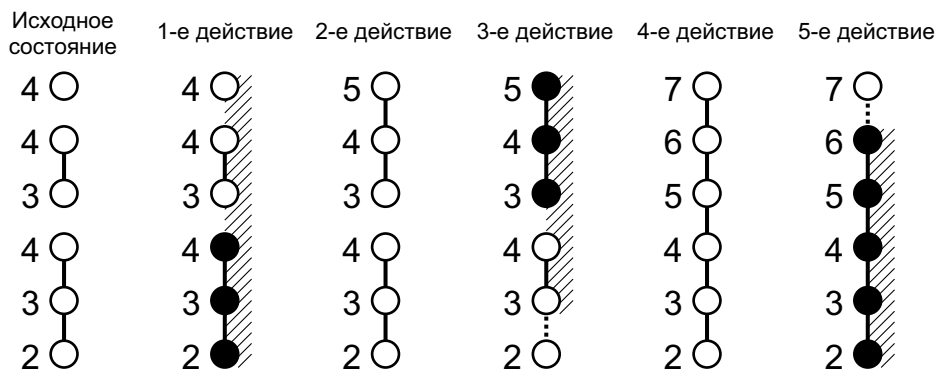
Для каждого действия второго типа выведите в новой строке одно число — длину наибольшей цепочки.

Примеры

| atoms.in | atoms.out |
|-------------|-----------|
| 6 | 3 |
| 2 3 4 3 4 4 | 3 |
| 5 | 5 |
| ? 1 6 | |
| + 6 6 1 | |
| ? 2 6 | |
| + 4 6 2 | |
| ? 1 5 | |

Замечание

Иллюстрация к примеру. Пунктиром выделены сильные связи, которые разрушаются на время действия второго типа. Для каждого действия второго типа выделены отрезок запроса и самая длинная цепочка.



Задача F. Совет стаи

| | |
|-------------------------|--------------|
| Имя входного файла: | council.in |
| Имя выходного файла: | council.out |
| Ограничение по времени: | 3 секунды |
| Ограничение по памяти: | 256 мегабайт |

— Смотрите хорошенько, смотрите хорошенько, о волки!

Акела

Закон Джунглей очень ясно говорит, что каждый вновь женившийся волк может отделиться от своей стаи; однако едва его волчата вырастут настолько, чтобы хорошо держаться на ногах, он обязан привести их и представить Совету стаи, который обыкновенно собирается в полнолуние; это делается для того, чтобы остальные волки узнали их. После такого осмотра волчата имеют право бегать куда им угодно и пока они не поймают первого оленя.

На совете стаи волки представляют своих детенышей другим волкам, чтобы они могли принять решение об их вступлении в стаю. Совет проходит в полночь, и волчата располагаются вдоль полоски лунного света, чтобы их можно было лучше рассмотреть.

Процесс выбора вступающих в стаю волчат не совсем определен, но зависит от решения каждого из волков. Волк, оценивающий потомство, как правило, не изучает всех прибывших детенышей, но ограничивается лишь некоторыми, стоящими подряд. Также, иногда та или другая волчица выталкивает носом своего детёныша в полосу лунного света, желая, чтобы его непременно заметили. В таком случае какой-то другой волчонок уходит в тень, а новый встает на его место.

Показателем будущих способностей маленького волка является его рост, главным образом зависящий от длины его ног. Взрослые волки внимательно рассматривают волчат и выбирают из них несколько рядом стоящих, по их мнению, наиболее приспособленных для охоты. Приспособленность детеныша волки определяют, как число $h - e$, где h — рост волчонка, а e — некоторая эталонная величина, передающаяся волками из поколения в поколение. Таким образом, среди кандидатов, рассматриваемых им, каждый волк выбирает подряд стоящих, так, чтобы сумма их приспособленностей была максимальна. При этом, каждый волк понимает важность преемственности и выбирает хотя бы одного волчонка, даже если все кандидаты неважные.

Предскажите выбор каждого из волков на совете стаи.

Формат входных данных

В первой строке входного файла записано одно число N ($1 \leq N \leq 100\,000$) — количество волчат-кандидатов, выстроившихся в ряд в начале процесса отбора. Во второй строке записаны N целых чисел x_i ($|x_i| \leq 10^9$) — приспособленности детенышей.

В третьей строке записано одно число M ($0 \leq M \leq 100\,000$) — количество событий, произошедших на совете. В следующих M строках описываются события:

- 1 $a_i x_i$ ($1 \leq a_i \leq N, |x_i| \leq 10^9$) означает, что мать-волчица поставила своего детеныша с приспособленностью x_i на позицию a_i ;
- 2 $l_i r_i$ ($1 \leq l_i \leq r_i \leq N$) означает, что волк производит выбор, просматривая детенышей на позициях $[l_i, r_i]$.

Формат выходных данных

Для каждого события типа 2 выведите ответ — максимальную возможную сумму приспособленностей расположенных подряд волчат на отрезке $[l_i, r_i]$.

Примеры

| council.in | council.out |
|------------|-------------|
| 3 | 6 |
| 1 2 3 | 3 |
| 5 | 3 |
| 2 1 3 | 1 |
| 1 2 -1 | |
| 2 1 3 | |
| 2 2 3 | |
| 2 1 2 | |