

## Задача А. Мощные юнги

Имя входного файла: `power.in`  
Имя выходного файла: `power.out`  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Имеется список из  $n$  юнг, для каждого из которых известен его рост  $a_1, a_2, \dots, a_n$ . Рассмотрим некоторый его подсписок  $a_l, a_{l+1}, \dots, a_r$ , где  $1 \leq l \leq r \leq n$ , и для каждого натурального числа  $s$  обозначим через  $K_s$  число юнг с ростом  $s$  в этом подсписке. Назовем *мощностью* подсписка сумму произведений  $K_s \cdot K_s \cdot s$  по всем различным натуральным  $s$ . Так как количество различных чисел в массиве конечно, сумма содержит лишь конечное число ненулевых слагаемых.

Необходимо вычислить мощности каждого из  $t$  заданных подсписков.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $t$  ( $1 \leq n, t \leq 200000$ ) — длина списка и количество запросов соответственно.

Вторая строка содержит  $n$  натуральных чисел  $a_i$  ( $1 \leq a_i \leq 10^6$ ) — рост юнг.

Следующие  $t$  строк содержат по два натуральных числа  $l$  и  $r$  ( $1 \leq l \leq r \leq n$ ) — индексы левого и правого концов соответствующего подсписка.

### Формат выходных данных

Выведите  $t$  строк, где  $i$ -ая строка содержит единственное натуральное число — мощность подсписка  $i$ -го запроса.

### Примеры

| <code>power.in</code> | <code>power.out</code> |
|-----------------------|------------------------|
| 3 2                   | 3                      |
| 1 2 1                 | 6                      |
| 1 2                   |                        |
| 1 3                   |                        |
| 8 3                   | 20                     |
| 1 1 2 2 1 3 1 1       | 20                     |
| 2 7                   | 20                     |
| 1 6                   |                        |
| 2 7                   |                        |

## Задача В. Варенье

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

Малыш и Карлсон решили пойти на прогулку. Они знают, что прогулка будет совсем скучной, если перед ней не опустошить несколько банок варенья.

Малыш достал из кладовки  $N$  банок варенья и выставил их в ряд. В банке номер  $i$  содержится ровно  $a_i$  грамм варенья. Карлсон немного подумал и решил, что в некоторых банках недостаточно варенья, и что в банке номер  $i$  должно быть хотя бы  $b_i$  грамм варенья.

Выходить из этой ситуации Карлсон хочет в  $M$  этапов. На каждом этапе он выбирает числа  $l$ ,  $r$ ,  $x$  и  $y$ , а затем выполняет следующие операции: в банку номер  $l$  он добавляет  $x$  грамм варенья, в банку номер  $l + 1 - x + y$  грамм варенья, в банку номер  $l + 2 - x + 2 \cdot y$ , и так далее. В банку номер  $r$  наш герой добавит  $x + y \cdot (r - l)$  грамм варенья.

Малышу хочется определить для каждой банки  $i$  наименьший номер операции, после которой в ней станет хотя бы  $b_i$  грамм варенья. Помогите Малышу: найдите соответствующее число для каждой банки.

### Формат входных данных

В первой строке входного файла задано одно число  $N$  ( $1 \leq n \leq 10^5$ ) — количество банок. Во второй строке заданы  $N$  чисел  $a_i$  ( $0 \leq a_i \leq 2 \cdot 10^9$ ) — изначальное количество варенья в банке номер  $i$ . В третьей строке заданы  $N$  чисел  $b_i$  ( $0 \leq b_i \leq 2 \cdot 10^9$ ) — минимальное количество варенья, которое должно быть в банке номер  $i$ .

В четвертой строке задано  $M$  ( $0 \leq M \leq 10^5$ ) — число этапов добавления варенья в банки, которые выполнит Карлсон. В следующих  $M$  строках описаны сами этапы в хронологическом порядке. Каждый этап задан четырьмя числами  $l$ ,  $r$ ,  $x$  и  $y$  ( $1 \leq l \leq r \leq N$ ,  $0 \leq x, y \leq 3 \cdot 10^5$ ).

### Формат выходных данных

Выведите  $N$  чисел в одной строке, разделенные пробелом. Число номер  $i$  должно быть равно нулю, если в банке номер  $i$  изначально было достаточно варенья, номеру этапа, после которого в ней станет хотя бы  $b_i$  варенья, или  $-1$ , если даже после выполнения всех этапов, в этой банке будет недостаточно варенья. Этапы нумеруются с единицы.

### Примеры

| jam.in    | jam.out    |
|-----------|------------|
| 5         | 1 2 0 3 -1 |
| 5 4 4 2 1 |            |
| 7 7 4 7 7 |            |
| 3         |            |
| 1 2 2 0   |            |
| 2 5 1 1   |            |
| 3 4 2 2   |            |

## Задача С. Инверсии на отрезке

Имя входного файла: `riq.in`  
Имя выходного файла: `riq.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 64 мегабайта

Дана перестановка из  $n$  элементов.

Ответьте на  $m$  запросов про число инверсий для подотрезка перестановки от  $l$  до  $r$ .

Инверсией называется пара индексов  $i, j$  такая, что  $i < j$  и  $a_i > a_j$ , где  $a_i$  - это  $i$ -ый элемент перестановки.

### Формат входных данных

В первой строке задано число  $n$  ( $1 \leq n \leq 10^5$ ).

Во второй строке задана перестановка из  $n$  элементов (элементы перестановки — попарно различные целые числа от 1 до  $n$ ).

В третьей строке задано число  $m$  ( $1 \leq m \leq 10^5$ ).

В последующих  $m$  строках содержится по два числа,  $l$  и  $r$ , границы запроса ( $1 \leq l, r \leq n$ ).

### Формат выходных данных

Выведите  $m$  строк, ответы на данные запросы.

### Примеры

| riq.in      | riq.out |
|-------------|---------|
| 5           | 2       |
| 4 5 2 3 1   | 2       |
| 3           | 8       |
| 1 3         |         |
| 3 5         |         |
| 1 5         |         |
| 6           | 1       |
| 5 2 4 3 1 6 | 4       |
| 4           | 8       |
| 4 6         | 8       |
| 2 5         |         |
| 1 5         |         |
| 1 6         |         |

## Задача D. 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 Пупы есть  $n$  строк  $S_1, S_2, \dots, S_n$ . Однажды его друг Луна пришёл в бухгалтерию и попросил его ответить на  $q$  вопросов: сколько строк среди  $S_{l_i}, S_{l_i+1}, \dots, S_{r_i}$  содержат  $P_i$  как подстроку?

Но в бухгалтерии все перепутали, и ответить на запросы за Пупу придется Вам.

### 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 Е. Фаброзавры-дизайнеры

Имя входного файла: `fabro.in`  
Имя выходного файла: `fabro.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 64 мегабайта

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 F. Ралли и штампы

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дан связный неориентированный граф с  $n$  вершинами и  $m$  рёбрами. Вершины пронумерованы от 1 до  $n$ , а рёбра пронумерованы от 1 до  $m$ , где  $i$ -е ребро соединяет вершины  $a_i$  и  $b_i$ .

На этом графе  $q$  команд из двух человек участвует в ралли со штампами. Ралли для  $i$ -й команды выглядит следующим образом:

- Один участник команды начинает в вершине  $x_i$ , другой — в вершине  $y_i$ .
- Оба из них исследуют граф, двигаясь по рёбрам. Им нужно посетить суммарно  $z_i$  вершин, включая их начальные вершины. При этом каждая вершина учитывается не более одного раза, даже если она посещалась несколько раз или посещалась обоими участниками команды.
- Оценка команды определяется как наибольший номер ребра, по которому кто-то из них прошёл. Их цель — минимизировать это значение.

Найти минимально возможный балл для каждой пары.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $3 \leq n \leq 10^5$ ,  $n - 1 \leq m \leq 10^5$ ) — количество вершин и рёбер в графе.

Каждая из следующих  $m$  строк содержит два целых числа  $a_i$  и  $b_i$  ( $1 \leq a_i < b_i \leq n$ ), задающие рёбра графа.

Следующая строка содержит одно целое число  $q$  ( $1 \leq q \leq 10^5$ ) — количество команд.

Каждая из следующих  $q$  строк содержит три целых числа  $x_i$ ,  $y_i$  и  $z_i$  ( $1 \leq x_i < y_i \leq n$ ,  $3 \leq z_i \leq n$ ), задающие начальные вершины для участников очередной команды и требуемое число вершин для посещения.

Гарантируется, что граф является связным.

### Формат выходных данных

Для каждой команды выведите одно целое число — их минимально возможный результат.

### Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 5 6              | 1                 |
| 2 3              | 2                 |
| 4 5              | 3                 |
| 1 2              | 1                 |
| 1 3              | 5                 |
| 1 4              | 5                 |
| 1 5              |                   |
| 6                |                   |
| 2 4 3            |                   |
| 2 4 4            |                   |
| 2 4 5            |                   |
| 1 3 3            |                   |
| 1 3 4            |                   |
| 1 3 5            |                   |

## Задача G. МЕХ на пути

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 6 секунд  
Ограничение по памяти: 256 мегабайт

Дано дерево, на каждом ребре которого написано неотрицательное целое число. Вам необходимо ответить на несколько запросов вида «для данных вершин  $u$ ,  $v$  назовите наименьшее неотрицательное целое число, которое **не** встречается среди чисел, написанных на ребрах на пути от  $u$  до  $v$ ».

### Формат входных данных

Первая строка входных данных содержит два числа  $n$  и  $q$  ( $2 \leq n \leq 10^5$ ,  $1 \leq q \leq 10^5$ ), количество вершин и количество запросов.

Следующие  $n - 1$  строк содержат по три числа  $u_i, v_i, x_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ,  $0 \leq x_i \leq 10^9$ ), которые описывают ребро дерева  $(u_i, v_i)$ , на котором написано число  $x_i$ .

Следующие  $q$  строк содержат по паре чисел  $a_j, b_j$  ( $1 \leq a_j, b_j \leq n$ ), которая обозначает запрос на пути от  $a_j$  до  $b_j$ .

### Формат выходных данных

Для каждого запроса выведите единственное число — минимальное число, которое не встречается на пути.

### Примеры

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

## Задача Н. Леха и сумма

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

Вероятно, вы уже поняли, что Леха очень любит различные типы запросов.

Иногда, когда он идет гулять с друзьями, они идут в кино. Но каждый раз, когда Леха куда-то идет, он не может расслабиться, если его домашняя работа еще не выполнена.

Леха любезно просит вас помочь ему с домашним заданием.

В этой задаче вам дан массив  $p$ , состоящий из  $n$  элементов.

Вам нужно ответить на  $m$  запросов.

Каждый запрос имеет следующий формат:  $l \ r \ x$ . Ответ на запрос —  $\sum_{i=l}^r (p_i \bmod x)$ .

Для каждого запроса выведите ответ на отдельной строке.

### Формат входных данных

В первой строке содержатся два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 2 \cdot 10^5$ ).

Во второй строке содержится  $n$  целых чисел  $p_i$  ( $0 \leq p_i \leq 2 \cdot 10^5$ ).

В следующих  $m$  строках содержится по три целых числа:  $l$ ,  $r$  и  $x$  ( $1 \leq l \leq r \leq n$ ,  $1 \leq x \leq 2 \cdot 10^5$ ).

### Формат выходных данных

Для каждого запроса выведите ответ на отдельной строке.

### Примеры

| стандартный ввод            | стандартный вывод |
|-----------------------------|-------------------|
| 10 5                        | 11                |
| 23 32 42 50 2 33 41 5 100 3 | 75                |
| 2 10 3                      | 36                |
| 1 9 23                      | 9                 |
| 3 10 12                     | 13                |
| 2 8 4                       |                   |
| 2 10 5                      |                   |

### Замечание

- $2 + 0 + 2 + 2 + 0 + 2 + 2 + 1 + 0 = 11$
- $0 + 9 + 19 + 4 + 2 + 10 + 18 + 5 + 8 = 75$
- $6 + 2 + 2 + 9 + 5 + 5 + 4 + 3 = 36$
- $0 + 2 + 2 + 2 + 1 + 1 + 1 = 9$
- $2 + 2 + 0 + 2 + 3 + 1 + 0 + 0 + 3 = 13$

## Задача I. Пересечение перестановок

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 15 секунд  
Ограничение по памяти: 512 мегабайт

Заданы две перестановки  $a$  и  $b$ , обе состоят из  $n$  элементов. Перестановка из  $n$  элементов — это такая последовательность целых чисел, что каждое число от 1 до  $n$  встречается ровно один раз.

Запрашивается два типа запросов к ним:

- 1  $l_a r_a l_b r_b$  — посчитать количество значений, которые встречаются как в отрезке  $[l_a; r_a]$  позиций перестановки  $a$ , так и в отрезке  $[l_b; r_b]$  позиций перестановки  $b$ ;
- 2  $x y$  — поменять местами значения на позициях  $x$  и  $y$  перестановки  $b$ .

Выведите ответы на все запросы первого типа.

Гарантируется, что во входных данных существует хотя бы один запрос первого типа.

### Формат входных данных

В первой строке записаны два целых числа  $n$  и  $m$  ( $2 \leq n \leq 2 \cdot 10^5$ ,  $1 \leq m \leq 2 \cdot 10^5$ ) — количество элементов в обеих перестановках и количество запросов.

Во второй строке записаны  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ) — перестановка  $a$ . Гарантируется, что каждое число от 1 до  $n$  содержится в  $a$  ровно один раз.

В третьей строке записаны  $n$  целых чисел  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq n$ ) — перестановка  $b$ . Гарантируется, что каждое число от 1 до  $n$  содержится в  $b$  ровно один раз.

Каждая из следующих  $m$  строк содержит описание определенного запроса. Они бывают следующих видов:

- 1  $l_a r_a l_b r_b$  ( $1 \leq l_a \leq r_a \leq n$ ,  $1 \leq l_b \leq r_b \leq n$ );
- 2  $x y$  ( $1 \leq x, y \leq n$ ,  $x \neq y$ ).

### Формат выходных данных

Выведите ответы на запросы первого типа, каждый ответ в отдельной строке — количество значений, которые встречаются и в отрезке  $[l_a; r_a]$  позиций перестановки  $a$ , и в отрезке  $[l_b; r_b]$  позиций перестановки  $b$ .

### Примеры

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

### Замечание

Рассмотрим первый запрос первого примера. Значения на позициях  $[1; 2]$  перестановки  $a$  — это  $[5, 1]$ , а значения на позициях  $[4; 5]$  перестановки  $b$  — это  $[1, 4]$ . Только значение 1 встречается в обоих отрезках.

После первой смены мест (второй запрос) перестановка  $b$  становится  $[2, 1, 3, 5, 4, 6]$ .

После второй смены мест (шестой запрос) перестановка  $b$  становится  $[5, 1, 3, 2, 4, 6]$ .

## Задача J. 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:

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  |         |

### Замечание

TL для Python 4 секунды

## Задача К. И снова сумма...

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

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

- $add(i)$  — добавить в множество  $S$  число  $i$  (если он там уже есть, то множество не меняется);
- $sum(l, r)$  — 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  $sum(l, r)$ .

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

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.

### Примеры

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