

Содержание

Must have	2
Задача 18А. Неявный Ключ [0.5, 256]	2
Задачи здорового человека	3
Задача 18В. Range Minimum Query [0.6, 256]	3
Задача 18С. Вперёд! [0.3, 256]	4
Задача 18D. Переворачивания [1.0, 256]	5
Задача 18Е. Машенька и её интерес [0.2, 256]	6
Для искателей острых ощущений	7
Задача 18F. Поднимайся и вращай [0.1, 256]	7
Задача 18G. Вставка ключевых значений [0.4, 256]	8
Задача 18H. Эх, дороги... [0.25, 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 18А. Неявный Ключ [0.5, 256]

Научитесь быстро делать две операции с массивом:

- `add i x` — добавить после i -го элемента x ($0 \leq i \leq n$)
- `del i` — удалить i -й элемент ($1 \leq i \leq n$)

Формат входных данных

На первой строке n_0 и m ($1 \leq n_0, m \leq 10^5$) — длина исходного массива и количество запросов. На второй строке n_0 целых чисел от 0 до $10^9 - 1$ — исходный массив. Далее m строк, содержащие запросы. Гарантируется, что запросы корректны: например, если просят удалить i -й элемент, он точно есть.

Формат выходных данных

Выведите конечное состояние массива. На первой строке количество элементов, на второй строке сам массив.

Примеры

stdin	stdout
3 4	3
1 2 3	9 2 8
del 3	
add 0 9	
add 3 8	
del 2	

Подсказка по решению

Базовое дерево по неявному ключу. В каждой вершине размер.

Задачи здорового человека

Задача 18B. Range Minimum Query [0.6, 256]

Компания *Giggle* открывает свой новый офис в Судиславле, и вы приглашены на собеседование. Ваша задача — решить поставленную задачу.

Вам нужно создать структуру данных, которая представляет из себя массив целых чисел. Изначально массив пуст. Вам нужно поддерживать две операции:

- запрос: «? i j » — возвращает минимальный элемент между i -ым и j -м, включительно;
- изменение: «+ i x » — добавить элемент x после i -го элемента списка. Если $i = 0$, то элемент добавляется в начало массива.

Конечно, эта структура должна быть достаточно хорошей.

Формат входных данных

Первая строка входного файла содержит единственное целое число n — число операций над массивом ($1 \leq n \leq 200\,000$). Следующие n строк описывают сами операции. Все операции добавления являются корректными. Все числа, хранящиеся в массиве, по модулю не превосходят 10^9 .

Формат выходных данных

Для каждой операции в отдельной строке выведите её результат.

Примеры

stdin	stdout
8	4
+ 0 5	3
+ 1 3	1
+ 1 4	
? 1 2	
+ 0 2	
? 2 4	
+ 4 1	
? 3 5	

Подсказка по решению

В вершине теперь нужно ещё и хранить то, что просят.

Задача 18С. Вперёд! [0.3, 256]

Капрал Дукар любит раздавать приказы своей роте. Самый любимый его приказ — “Вперёд!”. Капрал строит солдат в ряд и отдаёт некоторое количество приказов, каждый из них звучит так: “Рядовые с l_i по l_j — вперёд!”

Перед тем, как Дукар отдал первый приказ, солдаты были пронумерованы от 1 до n , слева направо. Услышав приказ “Рядовые с l_i по l_j — вперёд!”, солдаты, стоящие на местах с l_i по l_j включительно, продвигаются в начало ряда, в том же порядке, в котором были.

Например, если в какой-то момент солдаты стоят в порядке 1, 3, 6, 2, 5, 4, то после приказа “Рядовые с 2 по 3 — вперёд!”, порядок будет таким: 3, 6, 1, 2, 5, 4. А если потом Капрал вышлет вперёд солдат с 3 по 4, то порядок будет уже таким: 1, 2, 3, 6, 5, 4.

Вам дана последовательность из приказов Капрала. Найдите порядок, в котором будут стоять солдаты после исполнения всех приказов.

Формат входных данных

В первой строке входного файла указаны числа n и m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 100\,000$) — число солдат и число приказов. Следующие m строк содержат приказы в виде двух целых чисел: l_i и r_i ($1 \leq l_i \leq r_i \leq n$).

Формат выходных данных

Выведите в выходной файл n целых чисел — порядок, в котором будут стоять солдаты после исполнения всех приказов.

Пример

stdin	stdout
6 3	1 4 5 2 3 6
2 4	
3 5	
2 2	

Подсказка по решению

Берём уже написанное дерево по неявному ключу, используем Split/Merge.

Задача 18D. Переворачивания [1.0, 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

Отложенные операции. Высплнить отрезок, вверджить обратно.

Задача 18Е. Машенька и её интерес [0.2, 256]

Есть n мальчиков и девочка Маша. Изначально каждый мальчик стоит сам по себе и i -й мальчик с точки зрения Маши имеет интересность i . Девочка Маша хочет провести некоторый эксперимент, в течение которого каждый мальчик стоит в некоторой шеренге. Мальчики несговорчивые, участвовать в эксперименте не хотят, поэтому Маша собирается прибегнуть к математическому моделированию. Для этого ей нужно научиться быстро обрабатывать следующие запросы:

- `link(a, b)` – взять мальчиков с номерами a и b , если они стоят в разных шеренгах, то объединить шеренгу в одну: в начале шеренга мальчика a , затем шеренга мальчика b .
- `split(a, k)` – взять шеренгу, в которой стоит мальчик с номером a и разбить её на две: первые k мальчиков и все остальные. Если размер шеренги не больше k , ничего делать не нужно.
- `interest(a, x)` – сделать интересность мальчика a равной x (целое от 0 до 10^9).
- `sum(a)` – суммарная интересность мальчиков в шеренге, в которой стоит мальчик a .

Формат входных данных

В первой строке n ($1 \leq n \leq 100\,000$) – количество мальчиков и m ($1 \leq m \leq 250\,000$) – количество запросов. Далее m строк. Для понимания формата смотри пример. Мальчики нумеруются числами от 1 до n .

Формат выходных данных

Для каждого запроса «sum» на отдельной строке одно число – суммарная интересность.

Примеры

stdin	stdout
5 12	1
sum 1	10
interest 5 10	18
sum 5	37
interest 3 7	27
link 3 1	10
link 3 5	
sum 1	
interest 1 20	
sum 1	
split 1 2	
sum 3	
sum 5	

Подсказка по решению

Теперь нужно ещё хранить отца и уметь подниматься от вершины до корня.

Для искателей острых ощущений

Задача 18F. Поднимайся и вращай [0.1, 256]

Изначально у вас есть n чисел $1, 2, \dots, n$. Каждое живёт само по себе. Далее числа будут объединяться в массивы. Вам нужно реализовать структуру, данных, умеющую отвечать на несколько запросов, за $\mathcal{O}(\log n)$ каждый.

- $+ i j$ – взять массивы, в которых живут числа i и j , и объединить их в один массив именно в таком порядке.
- $! i k$ – взять массив, в котором живёт число i , и повернуть его на k влево.
- $- i k$ – взять массив, в котором живёт число i , и отрезать первые k элементов. Получится два новых массива.

Гарантируется, что запросы корректны. В первом i и j живут в разных массивах, во втором и третьем, длина массива, содержащего i , строго больше k .

Формат входных данных

На первой строке число элементов n ($2 \leq n \leq 100\,000$) и число запросов m ($1 \leq m \leq 100\,000$).
На следующих m строках сами запросы.

Формат выходных данных

После всех запросов нужно вывести получившиеся массивы. На первой строке выведите число массивов. Далее k массивов в формате «число элементов и сами элементы». Отсортируйте массив массивов перед выводом (массивы сравниваются лексикографически).

Пример

stdin	stdout
5 5	3
+ 1 2	1 2
+ 1 3	3 3 4 1
+ 1 4	1 5
! 3 1	
- 2 1	

Подсказка по решению

Вам нужно каждому элементу сопоставить `node*` и научиться подниматься от неё до корня соответствующего декартова дерева.

Задача 18G. Вставка ключевых значений [0.4, 256]

Вас наняла на работу компания MascoHard, чтобы вы разработали новую структуру данных для хранения целых ключевых значений.

Эта структура выглядит как массив A бесконечной длины, ячейки которого нумеруются с единицы. Изначально все ячейки пусты. Единственная операция, которую необходимо поддерживать — это операция $\text{Insert}(L, K)$, где L — положение в массиве, а K — некоторое положительное целое ключевое значение. Операция выполняется следующим образом:

- Если ячейка $A[L]$ пуста, то присвоить $A[L] := K$.
- Иначе выполнить $\text{Insert}(L+1, A[L])$, а затем присвоить $A[L] := K$.

По заданной последовательности из N целых чисел L_1, L_2, \dots, L_N вам необходимо вывести содержимое этого массива после выполнения следующей последовательности операций:

```
Insert(L1, 1)
Insert(L2, 2)
...
Insert(LN, N)
```

Формат входных данных

В первой строке входного файла содержится N (число операций Insert) и M (максимальный номер позиции, которую можно использовать в операции Insert). ($1 \leq N, M \leq 131\,072$).

В следующей строке даны N целых чисел L_i ($1 \leq L_i \leq M$).

Формат выходных данных

Выведите содержимое массива после выполнения данной последовательности операций Insert . На первой строке выведите W — номер последней несвободной позиции в массиве. Далее выведите W целых чисел — $A[1], A[2], \dots, A[W]$. Для пустых ячеек выводите нули.

Пример

stdin	stdout
5 4	6
3 3 4 1 3	4 0 5 2 3 1

Задача 18Н. Эх, дороги... [0.25, 256]

В многострадальном Тридесятom государстве опять готовится дорожная реформа. Впрочем, надо признать, дороги в этом государстве находятся в довольно плачевном состоянии. Так что реформа не повредит. Одна проблема — дорожникам не развернуться, поскольку в стране действует жесткий закон — из каждого города должно вести не более двух дорог. Все дороги в государстве двусторонние, то есть по ним разрешено движение в обоих направлениях (разумеется, разметка отсутствует). В результате реформы некоторые дороги будут строиться, а некоторые другие закрываться на бессрочный ремонт.

Петя работает диспетчером в службе грузоперевозок на дальние расстояния. В связи с предстоящими реформами, ему необходимо оперативно определять оптимальные маршруты между городами в условиях постоянно меняющейся дорожной ситуации. В силу большого количества пробок и сотрудников дорожной полиции в городах, критерием оптимальности маршрута считается количество промежуточных городов, которые необходимо проехать.

Помогите Пете по заданной последовательности сообщений об изменении структуры дорог и запросам об оптимальном способе проезда из одного города в другой, оперативно отвечать на запросы.

Формат входных данных

В первой строке входного файла заданы числа n — количество городов, m — количество дорог в начале реформы и q — количество сообщений об изменении дорожной структуры и запросов ($1 \leq n \leq 100\,000$, $0 \leq m \leq 100\,000$, $0 \leq q \leq 200\,000$). Следующие m строк содержат по два целых числа каждая — пары городов, соединенных дорогами перед реформой. Следующие q строк содержат по три элемента, разделенных пробелами. «+ i j » означает строительство дороги от города i до города j , «- i j » означает закрытие дороги от города i до города j , «? i j » означает запрос об оптимальном пути между городами i и j .

Гарантируется, что в начале и после каждого изменения никакие два города не соединены более чем одной дорогой, и из каждого города выходит не более двух дорог. Никакой город не соединяется дорогой сам с собой.

Формат выходных данных

На каждый запрос вида «? i j » выведите одно число — минимальное количество промежуточных городов на маршруте из города i в город j . Если проехать из i в j невозможно, выведите -1 .

Пример

stdin	stdout
5 4 6	0
1 2	-1
2 3	1
1 3	2
4 5	
? 1 2	
? 1 5	
- 2 3	
? 2 3	
+ 2 4	
? 1 5	

Подсказка по решению

Нужно увидеть массивы, которые мерджатся и сплитятся.