Задача A. Переворот

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

Дан массив. Надо научиться обрабатывать два типа запросов.

- `1 L R` - перевернуть отрезок [L, R]
- `2 L R` - найти минимум на отрезке [L, R]

Формат входного файла

Первая строка файла содержит два числа n, m. ($1 \leqslant n, m \leqslant 10^5$) Во второй строке находится n чисел a_i ($1 \leqslant a_i \leqslant 10^9$)- исходный массив. Остальные m строк содержат запросы, в формате описанном в условии. Для чисел L,R выполняется ограничение ($1 \leqslant L \leqslant R \leqslant n$).

Формат выходного файла

На каждый запрос типа 2, во входной файл выведите ответ на него, в отдельной строке.

Примеры

reverse.in	reverse.out
10 7 5 3 2 3 12 6 7 5 10 12 2 4 9 1 4 6 2 1 8 1 1 8 1 8 9 2 1 7 2 3 6	3 2 2 2

Note

Эту задачу нужно сдавать с помощью Splay Tree.

Задача B. Динамический Лес

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

Вам нужно научиться обрабатывать 3 типа запросов:

1. Добавить ребро в граф (`link`).
2. Удалить ребро из графа (`cut`).
3. По двум вершинам a и b вернуть длину пути между ними (или -1, если они лежат в разных компонентах связности) (`get`).

Изначально граф пустой (содержит N вершин, не содержит ребер). Гарантируется, что в любой момент времени граф является лесом. При добавлении ребра гарантируется, что его сейчас в графе нет. При удалении ребра гарантируется, что оно уже добавлено.

Формат входного файла

Числа N и M ($1 \leq N \leq 10^5 + 1$, $1 \leq M \leq 10^5$) — количество вершин в дереве и, соответственно, запросов. Далее M строк, в каждой строке команда (`link` или `cut`, или `get`) и 2 числа от 1 до N — номера вершин в запросе.

Формат выходного файла

В выходной файл для каждого запроса `get` выведите одно число — расстояние между вершинами, или -1, если они лежат в разных компонентах связности.

Примеры

linkcut.in	linkcut.out
3 7 get 1 2 link 1 2 get 1 2 cut 1 2 get 1 2 link 1 2 get 1 2	-1 1 -1 1
5 10 link 1 2 link 2 3 link 4 3 cut 3 4 get 1 2 get 1 3 get 1 4 get 2 3 get 2 4 get 3 4	1 2 -1 1 -1 -1

Задача C. Дерево

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

Рассмотрим корневое дерево. Изначально дерево состоит из одного лишь корня. На

сыновьях каждой вершины определён порядок слева направо. Допустимые операции с деревом таковы:

- Добавить в дерево лист.
- Удалить из дерева лист.
- Найти количество вершин на пути между двумя листьями.
- Найти количество вершин «под путём» между двумя листьями.

Множество вершин, лежащих «под путём» между листьями u и v, определяется следующим образом. Рассмотрим путь $u = w_0 - w_1 - \ldots - w_k = v$ между ними. Выделим в нём ту вершину w_c, в которой оба ребра пути идут в её сыновей. Пусть для определённости левое из этих рёбер приближает нас к вершине u, а правое — к вершине v. Тогда лежащими «под путём» объявляются следующие вершины:

- Все сыновья w_c, лежащие между w_{c-1} и w_{c+1}, а также все вершины их поддеревьев;
- Для $i = 1, 2, \ldots, c-1$ — все сыновья w_i, лежащие правее сына w_{i-1}, а также все вершины их поддеревьев;
- Для $i = c+1, c+2, \ldots, k-1$ — все сыновья w_i, лежащие левее сына w_{i+1}, а также все вершины их поддеревьев.

Напишите программу, которая по последовательности запросов перестраивает дерево согласно запросам на добавление и удаление вершин, а также вычисляет ответы на запросы о количестве вершин на пути и «под путём».

Формат входного файла

В первой строке ввода содержится одно целое число n — количество запросов ($0 \le n \le 300\,000$). Каждая из следующих n строк описывает один запрос. Возможные виды запросов таковы:

- `l` x — добавить новый лист как самого левого сына вершины x
- `r` x — добавить новый лист как самого правого сына вершины x
- `a` x y — добавить новый лист как сына вершины x, находящегося непосредственно справа от вершины y; все сыновья x, находившиеся до этого справа от y, после добавления оказываются правее новой вершины; гарантируется, что y является сыном x.
- `d` x удалить вершину x. Гарантируется, что в этот момент вершина x не удалена и является листом.
- `p` x y найти количество вершин на пути между x и y, включая сами эти вершины; гарантируется, что x и y являются листьями
- `q` x y найти количество вершин «под путём» между x и y, включая сами эти вершины; гарантируется, что x и y являются листьями.

Добавляемые вершины нумеруются с единицы в том порядке, в котором они добавляются в запросах. Корень дерева имеет номер 0 и листом ни в какой момент времени не считается.

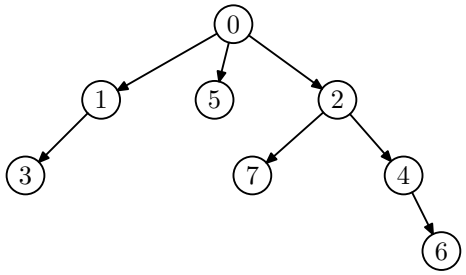
Формат выходного файла

Выполняя запросы по порядку, на каждый запрос вида `p` или `q` выведите ответ на него на отдельной строке.

Примеры

treenum.in	treenum.out
10 l 0 r 0 l 1 r 2 a 0 1 r 4 p 6 5 l 2 q 3 6 d 7	5 2

Note



На рисунке показано дерево до выполнения последнего запроса — удаления вершины 7.
Путь между вершинами 6 и 5 содержит пять вершин: 6 – 4 – 2 – 0 – 5.
«Под путём» между вершинами 3 и 6 находится две вершины — 5 и 7.
Эту задачу надо решать онлайн. Оффлайн решение получит Ignored.

Задача D. Соединение и разъединение

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

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” and “Посчитать количество компонент связности в графе” .

А вы когда-нибудь слышали о динамической задаче связности? В этой задаче вам надо обрабатывать три типа запросов:

1. Добавить ребро в граф
2. Удалить ребро из графа
3. Посчитать количество компонент связности в графе.

Формат входного файла

Изначально граф пустой.

В первой строке находятся два целых числа N и K — количество вершин и количество запросов сответственно ($1 \le N \le 300\,000$, $0 \le K \le 300\,000$). Следующие K строк содержат запросы, по одному в строке. Есть три типа запросов:

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 $u \neq v$. Граф считается неориентированным.

Формат выходного файла

Для каждого запроса типа ‘`?`’, Выведите количество компонент связности в момент запроса.

Примеры

connect.in	connect.out
5 11 ? + 1 2 + 2 3 + 3 4 + 4 5 + 5 1 ? - 2 3 ? - 4 5 ?	5 1 1 2

Задача E. Мосты: последняя битва

Имя входного файла:	`bridges3.in`
Имя выходного файла:	`bridges3.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

А у вашей дипломной работы есть практическое применение?

Популярный вопрос

Легенда

Мальчик Серёжа уже почти закончил работу над дипломной работой. Тема диплома с декабря не изменилась, это всё ещё «Dynamic 2-Edge-Connectivity Problem». Алгоритм уже придуман, протестирован, доказана оценка $O(K \log K)$... Осталось только написать главу про «практическое применение».

Ну какое может быть практическое применение у задачи «Dynamic 2-Edge-Connectivity Problem»? Вопрос непростой. Он оказался чуть ли не сложнее исходной задачи. Но диплом нужно скоро защитить, так что срочно нужно что-то делать.

Итак, первое практическое применение: можно предложить на соревнование задачу на эту тему!

Задача

Дан неориентированный граф из не более чем 10^5 вершин. Изначально граф не содержит рёбер. Вам нужно обрабатывать запросы вида `ADD x y` и `DEL x y` — добавить или удалить ребро из x в y.

В каждый момент времени нужно знать, **сколько в графе мостов**.

Кратных рёбер и петель ни в какой момент времени нет.

Если поступает команда «удалить ребро», оно в графе точно присутствует.

Решение задачи

Диплом на то и диплом, что за 5 часов его так просто не напишешь. Поэтому вам даны целых 5 подсказок.

1. Добавить ребро и удалить ребро — сделать ребро «живым» на некотором отрезке времени.
2. Используйте метод «Разделяй и Властвуй».
3. Сжимайте компоненты двусвязности.
4. Даже когда компоненты двусвязности уже сжаты, если запросов мало, граф всё ещё можно сильно уменьшить.
5. Существует решение за $O(K \log K)$.

Формат входного файла

В первой строке записаны целые числа N и K: количество вершин и количество запросов, соответственно. $1 \le N \le 10^5$, $1 \le K \le 10^5$.

В следующих K строках по одному в строке записаны запросы. Каждый запрос начинается словом «ADD» или «DEL» для запроса на добавление или удаление ребра, соответственно. После этого слова записаны два целых числа a и b, задающих ребро. $1 \le a, b \le N$, $a \ne b$.

Формат выходного файла

После каждого запроса нужно вывести одно число — сколько в текущем графе мостов.

Примеры

bridges3.in	bridges3.out
4 8 ADD 1 2 ADD 2 3 ADD 1 3 DEL 2 3 DEL 1 2 ADD 2 4 ADD 1 4 ADD 2 3	1 2 0 2 1 2 3 0

Задача F. Party

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

Byteasar intends to throw up a party. Naturally, he would like it to be a success. Furthermore, Byteasar is quite certain that to make it so it suffices if all invited guests know each other. He is currently trying to come up with a list of his friends he would like to invite.

Byteasar has n friends, where n is divisible by 3. Fortunately, most of Byteasar's friends know one another. Furthermore, Byteasar recalls that he once attended a party where there were $\frac{2}{3}n$ of his friends, and where everyone knew everyone else. Unfortunately, Byteasar does not quite remember anything else from that party... In particular, he has no idea which of his friends attended it.

Byteasar does not feel obliged to throw a huge party, but he would like to invite at least $\frac{n}{3}$ of his friends. He has no idea how to choose them, so he asks you for help.

Формат входного файла

In the first line of the input two integers, n and m ($3 \le n \le 1\,000$, $\frac{\frac{2}{3}n(\frac{2}{3}n-1)}{2} \le m \le \frac{n(n-1)}{2}$), are given, separated by a single space. These denote the number of Byteasar's friends and the number of pairs of his friends who know each other, respectively. Byteasar's friends are numbered from 1 to n. Each of the following m lines holds two integers separated by a single space. The numbers in line no. $i+1$ (for $i = 1, 2, \ldots, m$) are a_i and b_i ($1 \le a_i < b_i \le n$), separated by a single space, which denote that the persons a_i and b_i know each other. Every pair of numbers appears at most once on the input.

Формат выходного файла

In the first and only line of the output your program should print $\frac{n}{3}$ numbers, separated by single spaces, in increasing order. These number should specify the numbers of Byteasar's friends whom he should invite to the party. As there are multiple solutions, pick one arbitrarily.

Примеры

party.in	party.out
6 10 2 5 1 4 1 5 2 4 1 3 4 5 4 6 3 5 3 4 3 6	3 4

Задача G. Почтовая реформа

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

В Флатландии идет пора реформ. Недавно была проведена реформа дорог, так что теперь по дорогам страны из любого города можно добраться в любой другой, причем только одним способом. Также была проведена реформа волшебников, так что в каждом городе остался ровно один волшебник. Теперь же началась реформа почтовой системы.

Недавно образованное почтовое агентство «Экс-Федя» предлагает уникальную услугу — коллективную посылку. Эта услуга позволяет отправлять посылки жителям всех городов на каком-либо пути по цене обычной посылки. Удивительно, но пользоваться такой услугой стали только волшебники Флатландии, которые стали в большом количестве отправлять друг другу магические кактусы. Агентство столкнулось с непредвиденной проблемой: как известно, все волшебники живут в башнях и мало того, что не строят в них лестницы, так еще время от времени меняют их высоту. Поэтому, чтобы доставить посылку волшебнику, который живет в башне высотой h, курьеру агентства требуется иметь с собой не менее h метров веревки.

Вам поручено руководить отделом логистики — по имеющимся данным о высотах башен и об их изменениях вам нужно определять минимальную длину веревки, которую нужно выдать курьеру, который доставляет посылки между городами i и j.

Формат входного файла

Первая строка входного файла содержит число n — количество городов в Флатландии ($1 \le n \le 50\,000$). Во второй строке находится n положительных чисел, не превосходящих 10^5 — высоты башен в городах. В следующих $n-1$ строках содержится по два числа u_i и v_i — описание i-й дороги, $1 \le u_i, v_i \le n, u_i \ne v_i$. В следующий строке содержится число k — количество запросов ($1 \le k \le 100\,000$). В следующих k строках содержатся описания запросов в следующем формате:

- Уведомление от волшебника из города i о том, что высота его башни стала равна h, имеет вид ! i h, $1 \le i \le n$, $1 \le h \le 10^5$.
- Запрос от курьера о выдаче веревки для доставки посылок во все города на пути от i до j включительно имеет вид ? i j, $1 \le i, j \le n$.

Формат выходного файла

Для каждого запроса доставки посылок выведите минимальную длину веревки, которую необходимо выдать курьеру.

Примеры

mail.in	mail.out
3 1 2 3 1 3 2 3 5 ? 1 2 ! 1 5 ? 2 3 ! 3 2 ? 1 2	3 3 5
1 100 5 ! 1 1 ? 1 1 ! 1 1000 ? 1 1 ! 1 1	1 1000