

Задача А. LCA

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Дано подвешенное дерево с корнем в первой вершине. Вам нужно ответить на m запросов вида “найти LCA двух вершин”.

LCA вершин u и v в подвешенном дереве — это наиболее удалённая от корня дерева вершина, лежащая на обоих путях от u и v до корня.

Формат входных данных

В первой строке задано целое число n — число вершин в дереве ($1 \leq n \leq 2 \cdot 10^5$).

В следующих $n - 1$ строках записано одно целое число x . Число x на строке i означает, что x — предок вершины i ($x < i$).

Затем дано число m .

Далее заданы m ($0 \leq m \leq 5 \cdot 10^5$) запросов вида (u, v) — найти LCA двух вершин u и v ($1 \leq u, v \leq n$; $u \neq v$).

Формат выходных данных

Для каждого запроса выведите LCA двух вершин на отдельной строке.

Примеры

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

Задача В. Самое дешевое ребро

Имя входного файла: `minonpath.in`
Имя выходного файла: `minonpath.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дано подвешенное дерево с корнем в первой вершине. Все ребра имеют веса (стоимости). Вам нужно ответить на M запросов вида “найти у двух вершин минимум среди стоимостей ребер пути между ними”.

Формат входных данных

В первой строке задано целое число n — число вершин в дереве ($1 \leq n \leq 2 \cdot 10^5$).

В следующих $n - 1$ строках записаны два целых числа x и y . Число x на строке i означает, что x — предок вершины i , y задает стоимость ребра ($x < i$; $|y| \leq 10^6$).

Далее заданы m ($0 \leq m \leq 5 \cdot 10^5$) запросов вида (x, y) — найти минимум на пути из x в y ($1 \leq x, y \leq n$; $x \neq y$).

Формат выходных данных

Выведите ответы на запросы.

Примеры

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

Задача С. Прибавление на пути

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

Задано дерево. В каждой вершине есть значение, изначально все значения равны нулю. Требуется обработать запрос прибавления на пути и запрос значения в вершине.

Формат входных данных

В первой строке задано целое число n — число вершин в дереве ($1 \leq n \leq 3 \cdot 10^5$).

В следующих $n - 1$ строках заданы ребра дерева: по два целых числа v и u в строке — номера вершин, соединенных ребром ($1 \leq v, u \leq n$).

В следующей строке задано целое число m — число запросов ($1 \leq m \leq 5 \cdot 10^5$).

Следующие m строк содержат запросы в одном из двух форматов:

- `+ v u d` — прибавить число d во все значения в вершинах на пути от v до u ($1 \leq v, u \leq n$; $1 \leq d \leq 10^9$);
- `? v` — вывести значение в вершине v ($1 \leq v \leq n$).

Формат выходных данных

Выведите ответы на все запросы.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 5 | 1 |
| 1 2 | 3 |
| 1 3 | 1 |
| 3 4 | |
| 3 5 | |
| 5 | |
| + 2 5 1 | |
| ? 3 | |
| + 1 1 2 | |
| ? 1 | |
| ? 3 | |

Задача D. 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 входных данных

Первая строка входного файла содержит число n — количество городов в Флатландии ($1 \leq n \leq 50000$). 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, $1 \leq u_i, v_i \leq n, u_i \neq v_i$. В следующей строке содержится число k — количество запросов ($1 \leq k \leq 100000$). 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 $! i h$, $1 \leq i \leq n, 1 \leq h \leq 10^5$.
- Запрос от курьера о выдаче веревки для доставки посылок во все города на пути от i до j включительно имеет вид $? i j$, $1 \leq i, j \leq n$.

Формат выходных данных

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 |

Задача Е. Декомпозиция

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

Рассмотрим дерево T . Назовем деревом декомпозиции корневое дерево $D(T)$.

Выберем любую из вершин дерева T , назовем ее r . Рассмотрим все компоненты связности дерева T , после удаления вершины r : S_1, S_2, \dots, S_k . Тогда корнем $D(T)$ будет вершина r , а детьми r в $D(T)$ будут $D(S_1), D(S_2), \dots, D(S_k)$.

Вам задано T . Найдите дерево декомпозиции, высота которого не более 20. Высотой дерева называется максимальное число вершин, которые может содержать простой путь начинающийся в корне.

Формат входных данных

Первая строка содержит n — число вершин дерева T ($1 \leq n \leq 2 \cdot 10^5$).

Следующие $n - 1$ строк содержат ребра дерева. Каждое ребро описывается парой чисел v_i, u_i — концы ребра ($1 \leq v_i, u_i \leq n$).

Формат выходных данных

Выведите n чисел: i -е число — родитель вершины i в дереве декомпозиции, если вершина является корнем, выведите 0.

Примеры

| стандартный ввод | стандартный вывод |
|-----------------------------------------------------------|-------------------|
| 3 1 2 2 3 | 2 0 2 |
| 9 3 2 4 2 1 2 5 1 1 6 7 6 6 8 8 9 | 0 1 2 2 1 1 6 6 8 |

Задача F. Опекуны карнотавров

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Карнотавры очень внимательно относятся к заботе о своем потомстве. У каждого динозавра обязательно есть старший динозавр, который его опекает. В случае, если опекуна съедают (к сожалению, в юрский период такое не было редкостью), забота о его подопечных ложится на плечи того, кто опекал съеденного динозавра. Карнотавры — смертоносные хищники, поэтому их обычаи строго запрещают им драться между собой. Если у них возникает какой-то конфликт, то, чтобы решить его, они обращаются к кому-то из старших, которому доверяют, а доверяют они только тем, кто является их опекуном или опекуном их опекуна и так далее (назовем таких динозавров суперопекунами). Поэтому для того, чтобы решить спор двух карнотавров, нужно найти такого динозавра, который является суперопекуном для них обоих. Разумеется, беспокоить старших по пустякам не стоит, поэтому спорщики стараются найти самого младшего из динозавров, который удовлетворяет этому условию. Если у динозавра возник конфликт с его суперопекуном, то этот суперопекун сам решит проблему. Если у динозавра нелады с самим собой, он должен разобраться с этим самостоятельно, не беспокоя старших. Помогите динозаврам разрешить их споры.

Формат входных данных

Во входном файле записано число M , обозначающее количество запросов ($1 \leq M \leq 200000$). Далее на отдельных строках следуют M запросов, обозначающих следующие события:

- $+ v$ — родился новый динозавр и опекунство над ним взял динозавр с номером v . Родившемуся динозавру нужно присвоить наименьший натуральный номер, который до этого еще никогда не встречался.
- $- v$ — динозавра номер v съели
- $? u v$ — у динозавров с номерами u и v возник конфликт и вам надо найти им третейского судью.

Изначально есть один прадинозавр номер 1; гарантируется, что он никогда не будет съеден.

Формат выходных данных

Для каждого запроса типа «?» в выходной файл нужно вывести на отдельной строке одно число — номер самого молодого динозавра, который может выступить в роли третейского судьи.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 11 | 1 |
| + 1 | 1 |
| + 1 | 2 |
| + 2 | 2 |
| ? 2 3 | 5 |
| ? 1 3 | |
| ? 2 4 | |
| + 4 | |
| + 4 | |
| - 4 | |
| ? 5 6 | |
| ? 5 5 | |

Задача G. Генеалогия

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

Во время обсуждений в Парламенте лорды, с похожими взглядами на решение проблемы, обычно объединяются в группы. Как правило, результат обсуждения зависит от решения наиболее влиятельной группы лордов. Именно поэтому подсчёт влиятельности группы является наиболее важной задачей.

Естественно, каждый лорд дорожит древностью своего рода, поэтому влиятельность лорда равна древности его рода. Древность рода лорда — количество предков лорда: его отец, его дед, его прадед, и т.д. Чтобы посчитать влиятельность группы лордов, требуется посчитать количество лордов в группе вместе с их предками. Отметим, что если лорд является предком двух или более лордов в группе, то этот лорд должен быть посчитан только один раз.

Вам дано фамильное дерево лордов (удивительно, но все лорды произошли от одного пра-лорда) и список групп. Для каждой группы найдите её влиятельность.

Формат входных данных

Первая строка входного файла содержит число n — количество лордов ($1 \leq n \leq 100\,000$). Лорды нумеруются целыми числами от 1 до n . Следующая строка содержит n целых чисел p_1, p_2, \dots, p_n , где p_i — отец лорда с номером i . Если лорд является основателем рода, то p_i равно -1 . Гарантируется, что исходные данные формируют дерево. Третья строка входного файла содержит одно число g — количество групп ($1 \leq g \leq 3\,000\,000$). Следующие g строк содержат описания групп. j -ая строка содержит число k_j — размер j -ой группы, после которого следуют k_j различных чисел — номера лордов, состоящих в j -ой группе. Гарантируется, что сумма всех k_j во входном файле не превосходит $3\,000\,000$.

Формат выходных данных

В выходной файл выведите g строк. В j -ой строке выведите единственное число: влиятельность j -ой группы. Гарантируется, что размер выходного файла не превосходит шести мегабайт.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4 | 4 |
| -1 1 2 3 | 4 |
| 4 | 4 |
| 1 4 | 4 |
| 2 3 4 | |
| 3 2 3 4 | |
| 4 1 2 3 4 | |
| 5 | 4 |
| 2 -1 1 2 3 | 4 |
| 10 | 5 |
| 3 3 4 1 | 2 |
| 3 2 4 3 | 3 |
| 4 1 3 5 4 | 4 |
| 1 4 | 1 |
| 2 2 3 | 5 |
| 3 1 4 3 | 2 |
| 1 2 | 3 |
| 3 3 4 5 | |
| 1 1 | |
| 3 1 2 4 | |

Задача Н. Дерево

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

Задано подвешенное дерево, содержащее n ($1 \leq n \leq 1\,000\,000$) вершин. Каждая вершина покрашена в один из n цветов. Требуется для каждой вершины v вычислить количество различных цветов, встречающихся в поддереве с корнем v .

Формат входных данных

В первой строке входного файла задано число n . Последующие n строк описывают вершины, по одной в строке. Описание очередной вершины i имеет вид $p_i\ c_i$, где p_i — номер родителя вершины i , а c_i — цвет вершины i ($1 \leq c_i \leq n$). Для корня дерева $p_i = 0$.

Формат выходных данных

Выведите n чисел, обозначающих количества различных цветов в поддеревьях с корнями в вершинах $1, \dots, n$.

Пример

| стандартный ввод | стандартный вывод |
|--------------------------------------|-------------------|
| 5 2 1 3 2 0 3 3 3 2 1 | 1 2 3 1 1 |

Задача I. Размер компонент

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Есть граф из n вершин. Требуется обрабатывать следующие запросы:

- `link U V` — добавить ребро UV . Гарантируется, что до этого запроса вершины U и V были в разных компонентах связности.
- `cut U V` — удалить ребро UV . Гарантируется, что такое ребро существовало.
- `size V` — узнать размер компоненты связности вершины V .

Формат входных данных

Первая строка содержит два числа n ($2 \leq n \leq 10^5$) и m ($1 \leq m \leq 10^5$) — число вершин и число операций. Следующие m строк содержат операции.

Формат выходных данных

Для каждой операции `connected V U` выведите 1, если вершины в одной компоненте или 0 если в разных.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 5 10 | 3 |
| link 2 5 | 2 |
| link 1 5 | 1 |
| size 1 | 5 |
| cut 2 5 | |
| size 1 | |
| size 2 | |
| link 2 3 | |
| link 2 4 | |
| link 3 5 | |
| size 1 | |