

Задача А. Декомпозиция

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

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

decomposition.in	decomposition.out
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 В. Пути в дереве

Имя входного файла: `tree-paths.in`
Имя выходного файла: `tree-paths.out`
Ограничение по времени: 15 секунд
Ограничение по памяти: 512 мегабайт

Дано дерево из n вершин. Найдите для каждого d от 1 до $n - 1$ число путей длины d .

Формат входных данных

Первая строка содержит n — число вершин дерева ($1 \leq n \leq 50000$).

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 $n - 1$ число: i -е число — число путей длины i .

Примеры

<code>tree-paths.in</code>	<code>tree-paths.out</code>
3 1 2 2 3	2 1
9 3 2 4 2 1 2 5 1 1 6 7 6 6 8 8 9	8 10 10 6 2 0 0 0

Задача С. БДБД

Имя входного файла: `lwdb.in`
Имя выходного файла: `lwdb.out`
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 256 мегабайт

Большая Древесная База Данных создана для того, чтобы в ней можно было надежно сохранить и раскрасить любое дерево. В новой версии БДБД запланирован новый функционал, для реализации которого потребуются вновь переосмыслить теорию графов.

В БДБД хранится взвешенное дерево. В языке запросов Системы Управления Большой Древесной Базы Данных (СУБДБД) предусмотрены два вида запросов:

1. «1 v d c » — покрасить все вершины, находящиеся на расстоянии не более d от вершины v , в цвет c . Все вершины изначально окрашены в цвет с номером 0.
2. «2 v » — вывести цвет вершины v .

Необходимо запрограммировать работу СУБДБД и ответить на все запросы пользователя.

Формат входных данных

В первой строке число N ($1 \leq N \leq 10^5$) — количество вершин дерева.

Следующие $N - 1$ строк содержат описание ребер, по три числа в строке a_i, b_i, w_i ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$, $1 \leq w_i \leq 10^4$), где i -ое ребро имеет вес w_i и соединяет вершины a_i и b_i .

В следующей строке число Q ($1 \leq Q \leq 10^5$) — число запросов. В каждой из Q следующих строк запросы одного из двух видов:

1. Числа 1, v , d , c ($1 \leq v \leq N$, $0 \leq d \leq 10^9$, $0 \leq c \leq 10^9$).
2. Числа 2, v ($1 \leq v \leq N$).

Все числа во входных данных целые.

Формат выходных данных

Для каждого запроса второго типа необходимо вывести в отдельной строке цвет запрошенной вершины.

Примеры

lwdb.in	lwdb.out
5	6
1 2 30	6
1 3 50	0
3 4 70	5
3 5 60	7
8	
1 3 72 6	
2 5	
1 4 60 5	
2 3	
2 2	
1 2 144 7	
2 4	
2 5	

Задача D. Почтовая реформа

Имя входного файла:	mail.in
Имя выходного файла:	mail.out
Ограничение по времени:	4 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 ($1 \leq n \leq 50\,000$). 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, $1 \leq u_i, v_i \leq n, u_i \neq v_i$. В следующей строке содержится число k — количество запросов ($1 \leq k \leq 100\,000$). 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 $! i h, 1 \leq i \leq n, 1 \leq h \leq 10^5$.
- 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. Наименьший общий предок

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

У Бобо есть корневое дерево из n вершин, удобно пронумерованных числами $1, 2, \dots, n$. Вершина 1 — корень дерева, и i -я вершина имеет вес w_i .
Он хотел бы посчитать $f(2), f(3), \dots, f(n)$ где

$$f(i) = \sum_{j=1}^{i-1} w_{\text{LCA}(i,j)}.$$

Формат входных данных

Входные данные содержит ноль или более тестовых примеров и заканчиваются символом конца файла. Для каждого тестового примера:

Первая строка содержит число n ($2 \leq n \leq 2 \cdot 10^5$).

Вторая строка содержит n чисел w_1, w_2, \dots, w_n ($1 \leq w_i \leq 10^4$).

Третья строка содержит $(n - 1)$ чисел p_2, p_3, \dots, p_n , где p_i обозначает ребро из вершины p_i в вершину i ($1 \leq p_i \leq n$). Ребра образуют дерево.

Гарантируется, что сумма всех n не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого тестового примера, выведите $(n - 1)$ чисел: $f(2), f(3), \dots, f(n)$.

Примеры

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

Задача F. Адам и дерево

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

Когда у Адама появляется корневое дерево (связный неориентированный граф без циклов), он сразу начинает его раскрашивать. Более формально, каждому ребру он сопоставляет некоторый цвет таким образом, чтобы выполнялись два условия:

- Не существует вершины, у которой больше двух инцидентных ребер покрашены в один цвет.
- Для любых двух вершин, у которых есть инцидентные ребра, покрашенные в один цвет (скажем, c), путь между ними содержит ребра только цвета c .

Не все раскраски дерева нравятся Адаму одинаково. Рассмотрим путь от некоторой вершины до корня. Количество различных цветов на этом пути назовем стоимостью вершины. Стоимостью раскраски дерева будем называть максимальную стоимость среди всех вершин дерева. Помогите Адаму определить минимальную стоимость раскраски дерева.

Изначально дерево Адама состоит из одной вершины, которая имеет номер один и является корнем. За один ход Адам подвешивает к уже существующей вершине новую, которая получает номер, равный наименьшему положительному целому не занятому числу. После каждой операции вам нужно сообщать минимальную стоимость раскраски получившегося дерева.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 10^6$) — количество добавлений новых вершин. Во второй строке записано n чисел p_i ($1 \leq p_i \leq i$) — номера вершин, к которым подвешивают очередную вершину.

Формат выходных данных

Выведите n целых чисел — минимальные стоимости раскраски дерева после каждого добавления.

Примеры

стандартный ввод	стандартный вывод
11	1 1 1 1 1 2 2 2 2 3
1 1 1 3 4 4 7 3 7 6 6	

Замечание

На картинке изображен один из возможных вариантов раскраски дерева из примера в самый последний момент. Стоимость вершин с номерами 11 и 12 равна трем.

