

## 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<br>1 2<br>2 3   | 2 0 2             |
| 9<br>3 2<br>4 2<br>1 2<br>5 1<br>1 6<br>7 6<br>6 8<br>8 9 | 0 1 2 2 1 1 6 6 8 |

## Задача В. На далекой Амазонке

|                         |              |
|-------------------------|--------------|
| Имя входного файла:     | treeeg.in    |
| Имя выходного файла:    | treeeg.out   |
| Ограничение по времени: | 6.5 секунд   |
| Ограничение по памяти:  | 256 мегабайт |

В бассейне далёкой реки Амазонки расположены  $N$  городов, пронумерованных для удобства целыми числами от 1 до  $N$ . Всем известно, что местные леса непроходимы, и передвижение возможно только по рекам. Как следствие, схема соединения городов является деревом.

К несчастью, в этом году в бассейне далёкой Амазонки не на шутку разошлась эпидемия новой болезни — крабового гриппа. То и дело поступает информация о новых заболевших. Поначалу справляться с ней было легко, но вскоре почти все больницы были переполнены, и сейчас пациентов может принимать только госпиталь, находящийся в городе 1.

Для удобства граждан была открыта горячая линия, куда первым делом необходимо обратиться при появлении симптомов крабового (его ещё часто называют раковым) гриппа. Вам необходимо написать программу, которая будет отвечать на обращения пострадавших, учитывая при этом информацию о работающих больницах. Вам ещё повезло, что вы знаете все запросы заранее!

Более формально, поступают запросы трёх видов:

- «+  $v$ » — госпиталь города  $v$  снова может принимать больных. Гарантируется, что в момент перед этим запросом госпиталь города  $v$  не работал.
- «-  $v$ » — госпиталь города  $v$  не может больше принимать больных. Гарантируется, что в момент перед этим запросом госпиталь города  $v$  работал.
- «?  $v$ » — заболел человек в городе  $v$ , необходимо сообщить ему расстояние до ближайшего города с работающим госпиталем (в идеале неплохо бы ещё и сказать номер этого города, но этим пусть занимаются ваши коллеги). Гарантируется, что в момент такого запроса имеется хотя бы один работающий госпиталь.

### Формат входных данных

В первой строке находится единственное число  $N$  — количество городов ( $1 \leq N \leq 300\,000$ ). Следующие  $N - 1$  строк содержат информацию о соединениях между городами в формате « $u\ v\ l$ », что означает соединение между городами  $u$  и  $v$  длиной  $l$  километров ( $1 \leq u, v \leq N$ ,  $1 \leq l \leq 1000$ ). Направлением течения можно пренебречь и считать, что время движения зависит только от расстояний.

Далее на отдельной строке записано число  $Q$  — количество запросов ( $1 \leq Q \leq 300\,000$ ). Следующие  $Q$  строк содержат описание запросов в формате « $c\ v$ », где  $c$  — это один из трёх символов «+», «-» и «?», а  $v$  — номер города ( $1 \leq v \leq N$ ).

### Формат выходных данных

Для каждого запроса вида «?  $v$ » выведите на отдельной строке одно число — расстояние в километрах до ближайшего города с работающим госпиталем.

## Примеры

| treeeg.in | treeeg.out |
|-----------|------------|
| 5         | 6          |
| 1 2 2     | 4          |
| 2 3 3     | 7          |
| 3 4 1     |            |
| 3 5 4     |            |
| 5         |            |
| ? 4       |            |
| + 5       |            |
| ? 3       |            |
| - 1       |            |
| ? 2       |            |

## Задача С. Близкие вершины

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дано дерево из  $n$  вершин. Каждое ребро имеет неотрицательный вес. Длиной пути между двумя вершинами называется количество ребер в пути. Две вершины называются близкими, если существует путь между двумя этими вершинами длины не больше  $d$ . Посчитайте для каждой вершины  $u$  количество вершин  $v$ , таких, что  $u$  и  $v$  - близкие.

### Формат входных данных

В первой строке записаны три целых числа  $n, d$  ( $1 \leq n \leq 10^5, 0 \leq d \leq n$ ). Далее в  $n - 1$  строках дано описание ребер дерева. В  $i$ -й строке записано два целых числа  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ ), которые обозначают концы очередного ребра. Гарантируется, что рёбра задают дерево.

### Формат выходных данных

Выведите  $n$  целых чисел, где  $i$ -е число — количество близких пар с вершиной  $i$ .

### Примеры

| стандартный ввод                | стандартный вывод |
|---------------------------------|-------------------|
| 5 2<br>1 2<br>1 3<br>3 4<br>3 5 | 5 3 5 4 4         |
| 5 3<br>1 2<br>1 3<br>3 4<br>3 5 | 5 5 5 5 5         |
| 1 0                             | 1                 |

## Задача D. Гонки

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 1.5 секунд        |
| Ограничение по памяти:  | 256 мегабайт      |

Наряду с IOI в Паттайе проходят международные олимпийские гонки (IOR) 2011. Принимающей стороне требуется найти наиболее подходящую трассу для гонок.

В регионе Паттайя-Чонбури находятся  $N$  городов, соединённых сетью из  $(N - 1)$  магистралей. Каждая магистраль — двусторонняя, соединяет два различных города, и для нее известна длина в километрах — целое число. Известно, что между каждой парой городов существует ровно один возможный путь, соединяющий эти города. Таким образом, для любой пары городов существует ровно одна последовательность различных магистралей, по которой можно проехать из одного города в другой, не посещая никакой город дважды.

По требованиям организаторов IOR трасса должна являться путём суммарной длины *ровно*  $K$  километров, начинающимся и заканчивающимся в различных городах. Естественно, никакая магистраль и, поэтому, никакой город не могут быть использованы дважды при выборе трассы, иначе возможны столкновения. Чтобы минимизировать влияние гонок на трафик движения в регионе, необходимо выбрать для трассы путь из *наименьшего возможного количества магистралей*.

### Формат входных данных

В первой строке через пробел записаны пары чисел  $N$  и  $K$  — количество городов и требуемая длина трассы в километрах ( $1 \leq N \leq 200\,000$ ,  $1 \leq K \leq 1\,000\,000$ ). В следующих  $N - 1$  строках через пробел три целых числа  $u_i$ ,  $v_i$  и  $c_i$  — номера городов, соединённых магистралью, и длина этой магистрали, соответственно ( $0 \leq u_i, v_i \leq N - 1$ ,  $0 \leq c_i \leq 1\,000\,000$ ).

### Формат выходных данных

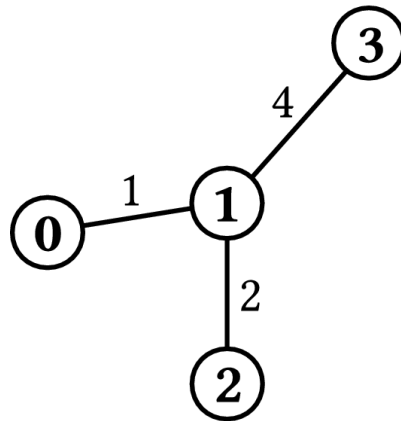
Выведите единственное число — минимальное возможное количество магистралей на допустимой трассе, имеющей длину, равную  $K$ . Если такой трассы не существует, выведите  $-1$ .

### Примеры

| стандартный ввод   | стандартный вывод |
|--|-------------------|
| 4 3<br>0 1 1<br>1 2 2<br>1 3 4   | 2                 |
| 3 3<br>0 1 1<br>1 2 1  | -1                |
| 11 12<br>0 1 3<br>0 2 4<br>2 3 5<br>3 4 4<br>4 5 6<br>0 6 3<br>6 7 2<br>6 8 5<br>8 9 6<br>8 10 7 | 2                 |

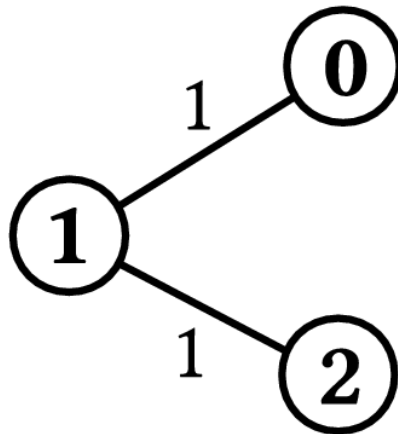
### Замечание

#### Пример 1



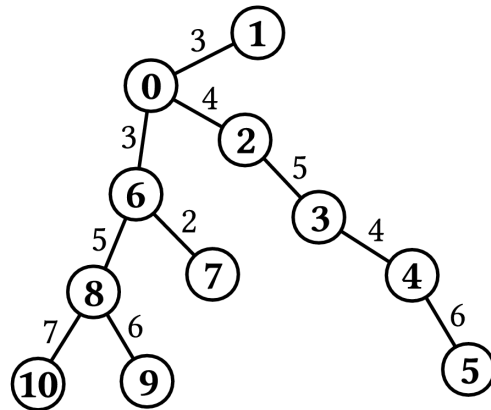
Единственная допустимая трасса начинается в городе 0, проходит через город 1 и заканчивается в городе 2.

**Пример 2**



В этом примере допустимой трассы не существует.

**Пример 3**



Одна из допустимых трасс состоит из 3 магистралей: она идёт из города с номером 6 через города с номерами 0 и 2 в город с номером 3. Другая трасса начинается в городе с номером 10 и идёт через город с номером 8 в город с номером 6. Вторая из них оптимальна, так как не существует подходящей трассы из одной магистрали.

## Задача Е. 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<br>1 2 3<br>1 3<br>2 3<br>5<br>? 1 2<br>! 1 5<br>? 2 3<br>! 3 2<br>? 1 2 | 3<br>3<br>5 |
| 1<br>100<br>5<br>! 1 1<br>? 1 1<br>! 1 1000<br>? 1 1<br>! 1 1              | 1<br>1000   |

## Задача F. 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 G. Котики и песики

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

Дима очень любит домашних животных. У него есть  $N$  домиков, пронумерованных от 1 до  $N$ . Они соединены  $N - 1$  дорожками, причем между любыми двумя домиками существует путь. Внутри каждого домика может быть не больше одного домашнего животного.

Дима хочет завести себе несколько котиков и песиков, но он обеспокоен тем, что они могут подражаться. Он определил **уровень опасности** как минимальное количество дорожек, которое нужно заблокировать, чтобы ни один песик и ни один котик не могли прийти друг до друга по незаблокированным дорожкам.

После определения уровня опасности Дима начал строить планы на ближайшие  $Q$  дней. Изначально, у него ни в одном домике нет животного. В  $i$ -й день план содержит одно из следующих действий:

- Завести котика и поселить его в домик с номером  $v$  (который до этого был пустым).
- Завести песика и поселить ее в домик с номером  $v$  (который до этого был пустым).
- Отдать питомца из домика с номером  $v$  в шаурмичную (это значит, что теперь домик с номером  $v$  пустой).

Дима хочет посчитать значение уровня опасности для каждого дня, но он слишком занят просмотром аниме. Поэтому вам надо посчитать эту величину за него.

### Формат входных данных

Первая строка содержит одно число  $N$  ( $1 \leq N \leq 100\,000$ ) — число домиков.

$(i + 1)$ -я строка содержит два числа  $v_i$  и  $u_i$ , обозначающие дорожку между двумя домиками  $v_i$  и  $u_i$ .

Следующая строка содержит одно число  $Q$  ( $1 \leq Q \leq 100\,000$ ) — число дней.

$(n + 1 + i)$ -я строка содержит информацию о планах Димы в следующей форме:

- 1  $v$  означает, что Дима заведет котика и поселит его в домике с номером  $v$  (который до этого был пустым).
- 2  $v$  означает, что Дима заведет собаку и поселит ее в домике с номером  $v$  (который до этого был пустым).
- 3  $v$  означает, что Дима отдаст своего питомца из домика с номером  $v$  в шаурмичную (это значит, что теперь домик с номером  $v$  пустой).

### Формат выходных данных

Выведите  $Q$  чисел — уровень опасности для каждого дня.

## Примеры

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