

## Задача А. 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 В. 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

<code>decomposition.in</code>	<code>decomposition.out</code>
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 С. Центроиды дерева

Имя входного файла: centroid.in  
Имя выходного файла: centroid.out  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Дано дерево из  $n$  вершин. У каждой вершины есть цвет. Нужно обработать  $q$  запросов  $(v_i, c_i)$ : найти расстояние от  $v_i$  до ближайшей к  $v_i$  вершины цвета  $c_i$ . Расстоянием между вершинами называется минимальное количество рёбер в пути между ними.

### Формат входных данных

На первой строке число  $n$  ( $1 \leq n \leq 10^5$ ), следующая строка содержит числа  $p_1, p_2, \dots, p_{n-1}$ .  $0 \leq p_i < i$ .  $p_i$  – отец вершины  $i$  в дереве. Далее строка с числами  $a_0, a_1, \dots, a_{n-1}$ .  $0 \leq a_i < n$ .  $a_i$  – цвет вершины  $i$ . Далее строка с числом  $q$  ( $1 \leq q \leq 10^5$ ). Следующие  $q$  строк содержат запросы  $v_i q_i$  ( $0 \leq v_i < n, 0 \leq c_i < n$ ).

### Формат выходных данных

Для каждого запроса выведите одно число – расстояние до ближайшей вершины нужного цвета, или  $-1$ , если в дереве нет вершин такого цвета.

### Примеры

centroid.in	centroid.out
5	0 1 2 -1 2 1 2 1 1
0 1 1 3	
1 2 3 2 1	
9	
0 1	
0 2	
0 3	
1 0	
2 1	
2 2	
3 3	
3 1	
4 2	

## Задача D. Дорешивание

Имя входного файла: `upsolving.in`  
Имя выходного файла: `upsolving.out`  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

В Летней Компьютерной Школе есть  $n$  параллелей, каждая из которых живёт в своём домике. Все параллели пронумерованы от 1 до  $n$  от младших к старшим. Периодически школьник, дорешивающий прошедшие практики у себя в домике, не справляется с задачей и идёт за помощью к товарищам из более старшей параллели.

Некоторые пары домиков соединены тропинками, всего есть  $n - 1$  такая тропинка. Все тропинки имеют одинаковую длину, по тропинке можно ходить между двумя домиками, которые она соединяет, и только между ними. От любого домика можно дойти до любого другого домика, используя только данные тропинки.

Если у школьника из параллели  $k$  не получается решить задачу, он из своего домика с номером  $k$  идёт просить помощи до какого-нибудь домика с номером, большим  $k$ . Поскольку ему не хочется тратить ни секунды драгоценного времени, он выбирает ближайший подходящий домик. Школьники из параллели  $n$  всегда решают свои задачи сами, так как им не к кому обратиться.

Вам дано описание тропинок между домиками. Для каждого  $k$  от 1 до  $n - 1$  определите минимальное расстояние, которое школьник из параллели  $k$  пройдёт в случае проблем с решением задачи.

### Формат входных данных

Первая строка входных данных содержит целое число  $n$  ( $1 \leq n \leq 200\,000$ ) — количество параллелей в ЛКШ.

В  $i$ -й из следующих  $n - 1$  строк содержатся два целых числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ,  $a_i \neq b_i$ ) — номера домиков, которые соединяет  $i$ -я тропинка.

Гарантируется, что каждую пару домиков соединяет не более одной тропинки, и что из любого домика можно дойти до любого другого.

### Формат выходных данных

Выведите  $n - 1$  строку,  $i$ -я из них должна содержать целое число  $d_i$  — расстояние до ближайшего домика с номером, большим  $i$ , от домика параллели  $i$ .

### Примеры

<code>upsolving.in</code>	<code>upsolving.out</code>
5	1
1 4	1
5 2	2
3 1	3
1 2	
5	1
4 3	2
3 5	1
5 1	2
1 2	

## Задача Е. 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 справиться 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. 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 F. Логистические вопросы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

*Огромное спасибо Михаилу MikeMirzayanov Мирзянову за невероятные системы Polygon и CodeForces, откуда и взята была эта замечательная задача.*

Некоторая страна состоит из  $n$  городов, соединённых железнодорожной сетью. Транспортное сообщение страны настолько развито, что сеть состоит из минимального необходимого количества двусторонних дорог в  $(n - 1)$  дорог (иными словами, граф дорог представляет собой дерево). Из них  $i$ -я дорога, непосредственно соединяющая города  $a_i$  и  $b_i$ , имеет длину  $l_i$  километров.

Транспортную систему обслуживает государственный перевозчик ХЖД (Хорошие Железные Дороги), который, в целях упрощения ценовой политики, предлагает единственный тариф для проезда на электричке. Для того, чтобы проехать маршрут длиной  $t$  километров, требуется заплатить  $t^{\frac{3}{2}}$  бурлей. При этом запрещается разбивать длинный маршрут на короткие отрезки и оплачивать их отдельно (за исполнением этого странного закона следит специальная железнодорожная полиция, ЖДП).

Большая Софтверная Компания решила организовать свой турнир по программированию. Проведя несколько отборочных туров, работники компании определили список финалистов и передали его в логистический отдел для подбора места проведения финала и решения вопроса закупки билетов. Большой Софтверной Компании не составит труда организовать финал турнира в любом из  $n$  городов страны, поэтому решающим фактором при выборе города для заключительного этапа соревнования является суммарная стоимость покупки билетов для всех финалистов. Известно, что в  $i$ -м городе страны живут  $w_i$  финалистов кубка.

Помогите сотрудникам компании определить город, суммарная стоимость проезда всех участников до которого является наименьшей.

### Формат входных данных

В первой строке входных данных находится число  $n$  ( $1 \leq n \leq 200\,000$ ) — количество городов в стране.

В следующей строке находятся  $n$  целых чисел  $w_1, w_2, \dots, w_n$  ( $0 \leq w_i \leq 10^8$ ) — количество финалистов, проживающих в каждом из городов страны.

В последующих  $(n - 1)$  строках находятся описания участков железной дороги,  $i$ -я строка содержит три целых числа  $a_i, b_i, l_i$  ( $1 \leq a_i, b_i \leq n, a_i \neq b_i, 1 \leq l_i \leq 1000$ ).

### Формат выходных данных

Выведите два числа — номер  $f$  оптимального города для проведения соревнования и вещественное число  $s$ , равное минимальной суммарной стоимости проезда всех финалистов до соревнования. Ваш ответ будет признан правильным, если одновременно выполнено два условия:

1. Абсолютная или относительная ошибка выведенного вами числа  $s$  относительно стоимости проведения города финала в  $f$  не превосходит  $10^{-6}$ ;
2. Абсолютная или относительная ошибка выведенного вами числа  $s$  относительно ответа жюри не превосходит  $10^{-6}$ .

Если правильных ответов несколько, разрешается вывести любой.

## Примеры

стандартный ввод	стандартный вывод
5 3 1 2 6 5 1 2 3 2 3 1 4 3 9 5 3 1	3 192.000000000000000000
2 5 5 1 2 2	1 14.142135623730951000

## Замечание

В тесте из условия оптимальный вариант выбора города для проведения финала соревнования — 3. При таком выборе стоимость проведения составит  $3 \cdot 4^{\frac{3}{2}} + 1 \cdot 1^{\frac{3}{2}} + 2 \cdot 0^{\frac{3}{2}} + 6 \cdot 9^{\frac{3}{2}} + 5 \cdot 1^{\frac{3}{2}} = 192$  бурлей.

Во втором тесте из условия вне зависимости от выбора города потребуется заплатить за проезд пяти участникам, на каждого будет потрачено по  $2\sqrt{2}$  бурлей.