

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. 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

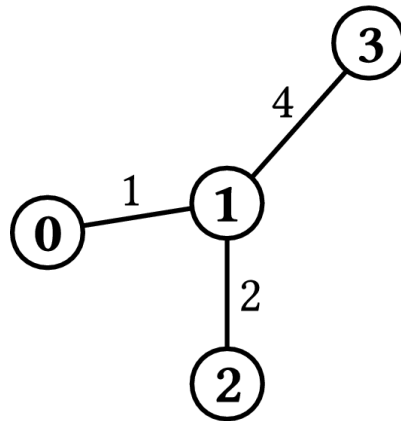
Выведите единственное число — минимальное возможное количество магистралей на допустимой трассе, имеющей длину, равную K . Если такой трассы не существует, выведите -1 .

Примеры

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

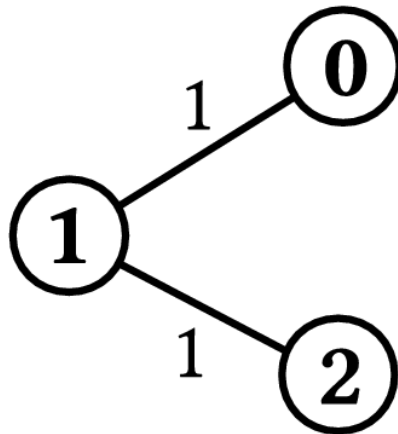
Замечание

Пример 1



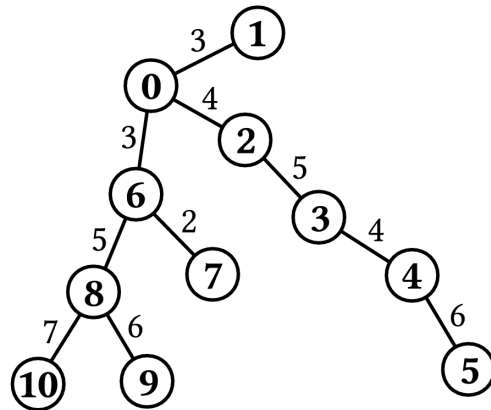
Единственная допустимая трасса начинается в городе 0, проходит через город 1 и заканчивается в городе 2.

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 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. 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 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 ($1 \leq Q \leq 300\,000$). 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 F. 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 домами, 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 домами. 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 G. Новые кампусы!

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	1024 мегабайта

Недавно вы стали ректором одного из университетов и решили открыть в нем новую программу! На ней вы будете учить студентов спортивному программированию. Поэтому у них будут два типа занятий: спорт (чтобы развить силу рук) и программирование. Основным достоинством этой программы будет обучение в двух кампусах одновременно — по четным дням студенты будут ездить в первый кампус, а по нечетным — во второй.

Оба кампуса вашего университета устроены очень необычно: в каждом из них есть по n аудиторий, пронумерованных от 1 до n , и по $n - 1$ переходу между ними, при этом из любой аудитории можно добраться в любую другую по переходам.

Однако вы обнаружили, что студентам сложно ориентироваться сразу в двух кампусах, и решили упростить им жизнь. Вы решили выбрать два номера аудиторий u и v ($u \neq v$): в аудитории с номером u студенты будут заниматься спортом, а в аудитории с номером v — программированием. Обратите внимание, что u и v выбираются одинаковыми для обоих кампусов.

Так как вы хотите, чтобы студенты тратили меньше времени на перемещение между аудиториями, вам нужно минимизировать суммарное расстояние, которое потребуется преодолеть студентам между выбранными аудиториями в каждом из кампусов. Более формально, вам нужно найти такие номера u, v , что $d_1(u, v) + d_2(u, v)$ минимально, где $d_1(u, v)$ — это расстояние между аудиториями u и v в первом кампусе, а $d_2(u, v)$ — во втором. Расстоянием между аудиториями называется минимальное число переходов, через которые нужно пройти, чтобы добраться из одной аудитории в другую.

В обоих кампусах есть вход, и он ведет в аудиторию 1. Для всех остальных аудиторий разработан план эвакуации. В первом кампусе для i -й аудитории p_i равно номеру следующей аудитории на пути из i -й аудитории в первую. Во втором кампусе для i -й аудитории q_i равно номеру следующей аудитории на пути из i -й аудитории в первую.

Формат входных данных

В первой строке входных данных содержится одно целое число n ($2 \leq n \leq 3 \cdot 10^5$) — количество аудиторий.

В следующей строке находятся $n - 1$ целых чисел $p_2, p_3, p_4, \dots, p_n$ ($1 \leq p_i \leq n$), где p_i — это следующая (кроме i) аудитория на пути от i -й до первой в первом кампусе.

В следующей строке находятся $n - 1$ целых чисел $q_2, q_3, q_4, \dots, q_n$ ($1 \leq q_i \leq n$), где q_i — это следующая (кроме i) аудитория на пути от i -й до первой во втором кампусе.

Формат выходных данных

В первой строке выведите минимальную величину $d_1(u, v) + d_2(u, v)$.

Во второй строке выведите любую пару вершин u, v , таких что $d_1(u, v) + d_2(u, v)$ минимально.

Примеры

стандартный ввод	стандартный вывод
5 3 1 2 3 5 4 1 3	2 5 3
5 5 1 2 3 4 4 1 4	2 2 4
7 1 2 2 7 1 3 5 5 5 1 5 2	3 2 1
9 5 2 1 4 9 8 3 7 1 4 7 9 8 2 5 3	4 2 1

Замечание

В первом примере в первом кампусе есть переходы между аудиториями 3 и 2, 1 и 3, 2 и 4, 3 и 5.
Во втором кампусе есть переходы между аудиториями 5 и 2, 4 и 3, 1 и 4, 3 и 5.