

Задача А. Командир Ciel

Имя входного файла: `commander.in`
Имя выходного файла: `commander.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Лиса Ciel становится командиром Древоземелья. В Древоземелье, как это следует из названия, есть n городов, соединенных $n - 1$ ненаправленными дорогами, а между любыми двумя городами существует путь по дорогам Древоземелья.

Лиса Ciel должна назначить каждому городу офицера. У каждого офицера есть ранг — буква от 'A' до 'Z'. Таким образом, имеется 26 различных рангов, самый высокий — 'A', самый низкий — 'Z'.

У Ciel имеется достаточно офицеров каждого ранга. Но не все так просто, должно быть выполнено особое условие: если x, y — два различных города и у их офицеров одинаковые ранги, то на простом пути между x и y должен быть город z , имеющий офицера более высокого ранга. Таким образом, общение между офицерами одного ранга будет гарантированно проходить под присмотром офицера более высокого ранга.

Помогите Ciel составить подходящий план назначения офицеров городам. Если это невозможно, выведите «Impossible!».

Формат входных данных

В первой строке записано целое число n ($2 \leq n \leq 10^5$) — количество городов в Древоземелье.

В каждой из следующих $n - 1$ строк записано два целых числа a и b ($1 \leq a, b \leq n, a \neq b$) — это значит, что существует дорога между городами a и b . Считайте, что города пронумерованы от 1 до n некоторым образом.

Гарантируется, что заданный граф будет деревом.

Формат выходных данных

Если подходящий план существует, выведите n символов, разделенных пробелами — i -ый символ обозначает ранг офицера в городе i .

В противном случае, выведите «Impossible!».

Примеры

<code>commander.in</code>	<code>commander.out</code>
4 1 2 1 3 1 4	A B B B

Замечание

В первом примере для любых двух офицеров ранга 'B', офицер с рангом 'A' будет на пути между ними. То есть, такое решение подходит.

Задача В. 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 С. 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

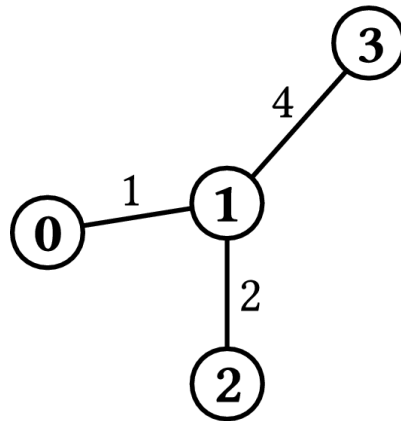
Выведите единственное число — минимальное возможное количество магистралей на допустимой трассе, имеющей длину, равную K . Если такой трассы не существует, выведите -1 .

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

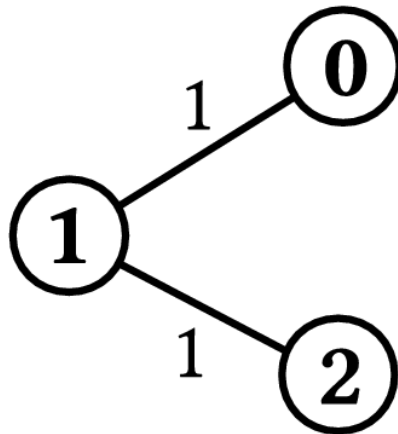
Замечание

Пример 1



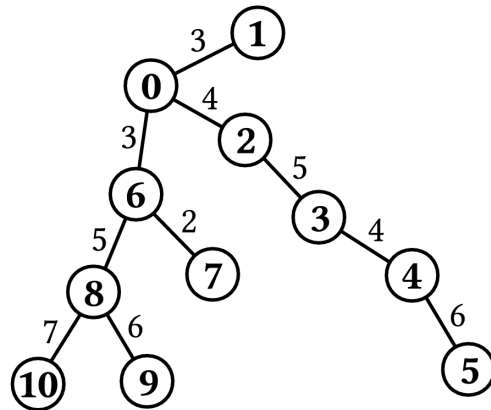
Единственная допустимая трасса начинается в городе 0, проходит через город 1 и заканчивается в городе 2.

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 D. В бухгалтерии опять всё перепутали

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

Лула и Пула пошли получать зарплату. Но в бухгалтерии опять всё перепутали. Лула получил зарплату за Пулу, а Пула . . .

Пула не хочет получать за Луну и хочет доказать бухгалтерии, что она не права.

Пула работает в крупной компании «MST Inc.», занимающейся информационным сопровождением «Всеберляндской олимпиады школьников по информатике». В компании «MST Inc.» работает n сотрудников, причём у каждого из них, кроме самой «MST», есть ровно один непосредственный начальник и несколько (возможно ноль) непосредственных подчинённых.

Всеми начальниками сотрудника компании «MST Inc.» называется множество, состоящее из его непосредственного начальника и множества начальников его непосредственного начальника. Известно, что у каждого сотрудника кроме самой «MST», «MST» входит в множество начальников этого сотрудника.

Множеством подчинённых у сотрудника называется множество, состоящее из него самого и множеств подчинённых у всех непосредственных подчинённых данного сотрудника. В частности, все сотрудники входят в множество подчинённых у «MST».

Каждый месяц каждому сотруднику начисляется зарплата, причём немаленькая, ведь иначе ни один сотрудник не согласился бы работать с «MST». Известно, что в нулевой месяц работы организации, каждому сотруднику заплатили по c_i бурлей. В качестве поощрения сотрудников «MST» придумала следующее правило: В каждый из следующих m месяцев берётся сотрудник с номером a_i и берётся число s_i — сумма зарплат всех сотрудников во множестве его начальников и подчинённых (включая его самого). Если это число оказывалось слишком большим, s_i берётся по модулю $10^9 + 7$. После этого берётся сотрудник с номером b_i , и к зарплате всех сотрудников, входящих во множество его начальников и подчинённых (включая его самого) прибавляется число s_i . С учётом этого изменения платится зарплата в i -й месяц и пересчитывается зарплата в следующие месяцы.

Вернёмся к Пуле. Пула хочет показать бухгалтерии компании «MST Inc.» что она всё перепутала, а для этого ему надо узнать, сколько же ему должны были заплатить в каждый из месяцев с нулевого по m -й. К сожалению, в гениальной системе поощрения, разработанной «MST», не может разобраться никто. Поэтому эту задачу поручили вам.

Формат входных данных

В первой строке входных данных даны 2 числа n и m ($1 \leq n, m \leq 10^5$) — число сотрудников компании «MST Inc.» и последний день, когда выплачивалась зарплата Пуле.

Во второй строке записано $n - 1$ число. i -е из них — номер непосредственного начальника сотрудника номер i (i принимает значения от 1 до $n - 1$). При этом «MST» имеет номер 0 и не имеет непосредственного начальника. Пула имеет номер $n - 1$.

В третьей строке записано n чисел c_i ($1 \leq c_i \leq 10^9$) — зарплата i -го сотрудника в нулевой день.

В каждой из следующих m строк записано по 2 числа a_i и b_i ($0 \leq a_i, b_i \leq n - 1$) — номер человека, на основе которого происходит поощрение и номер человека, к подчинённым и начальникам которого поощрение применяется (более подробно описано в условии).

Формат выходных данных

В единственной строке выведите $m + 1$ число — зарплату Пулы в каждый из дней с 0-го по m -й. Напоминаем, что Пула имеет номер $n - 1$. Обратите внимание, что зарплата **не считается** по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 3 0 0 1 1 1 0 0 2 1 1 2	1 4 4 28
4 3 0 1 1 0 1 0 0 0 1 1 3 2 3	0 1 6 20

Замечание

Пояснение к первому примеру:

В первый день к зарплате каждого сотрудника прибавилось 3 бурля и зарплаты стали соответственно 4, 4, 4.

Во второй день к зарплате сотрудников с номерами 0, 1 прибавилось по 8 бурлей и зарплаты стали соответственно 12, 12, 4.

Во третий день к зарплате сотрудников с номерами 0, 2 прибавилось по 24 бурля и зарплаты стали соответственно 36, 12, 28.

Задача E. 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 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 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 G. 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 Н. Близкие вершины

Имя входного файла: `close-vertices.in`
Имя выходного файла: `close-vertices.out`
Ограничение по времени: 5 секунд
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. Весом пути называется суммарный вес всех входящих в него ребер.

Две вершины называются близкими, если существует путь между двумя этими вершинами длины не более l и также существует путь между ними веса не более w . Определите количество близких пар вершин.

Формат входных данных

В первой строке записаны три целых числа n , l и w ($1 \leq n \leq 10^5, 1 \leq l \leq n, 0 \leq w \leq 10^9$). Далее в $n - 1$ строках дано описание ребер дерева. В i -той строке записано два целых числа p_i, w_i ($1 \leq p_i < (i + 1), 0 \leq w_i \leq 10^4$), которые обозначают, что i -ое ребро соединяет вершину $(i + 1)$ и p_i и имеет вес w_i .

Считайте, что вершины дерева пронумерованы от 1 до n некоторым образом.

Формат выходных данных

Выведите единственное целое число — количество близких пар.

Примеры

<code>close-vertices.in</code>	<code>close-vertices.out</code>
4 4 6 1 3 1 4 1 3	4

Задача I. Нет монет

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

Древляндия — страна из n городов, соединённых $n - 1$ дорогой так, что любые два города достижимы друг из друга по дорогам (возможно, не напрямую). Есть m запросов на транспортировку товаров, i -й запрос представляет собой доставку между городами u_i, v_i и даёт p_i монет.

Вы — маленькая лошадка, у которой пока что нет монет. Но вы очень хотите их заработать. Для этого вы хотите выбрать два города x и y и осуществить все транспортировки, у которых конечные города лежат на кратчайшем пути между x и y . Конечно, вы хотите выбрать x и y так, чтобы заработать как можно больше монет.

Формат входных данных

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

Следующие $n - 1$ строк содержат описание дорог. i -я из этих строк содержит два целых числа a_i, b_i ($1 \leq a_i, b_i \leq n$) — города, которые соединяет i -я дорога.

Вторая строка содержит целое число m ($0 \leq m \leq 10^5$).

Следующие m строк содержат описание запросов на транспортировку. i -я из этих строк содержит три целых числа u_i, v_i, p_i ($1 \leq u_i, v_i \leq n, 1 \leq p_i \leq 10^3$).

Формат выходных данных

Выведите единственное число — максимальное количество монет, которое вы можете получить.

Пример

стандартный ввод	стандартный вывод
6 1 2 2 3 2 4 5 4 6 4 4 1 4 10 2 5 20 6 3 15 2 1 1	31

Замечание

В примере оптимально выбрать $x = 1$ и $y = 5$.