Задача A. Телепорт

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

В королевстве есть n городов, пронумерованных числами от 1 до n. Город с номером 1 является столицей королевства.

В каждом городе построен телепорт — механизм, который может мгновенно переместить человека в другое место. Телепорты устроены таким образом, что из города i можно переместиться в город a_i. При этом гарантируется, что из любого города можно добраться до столицы, воспользовавшись телепортами некоторое количество раз.

Королю очень нравится число k, поэтому он хочет изменить направление некоторых телепортов таким образом, чтобы выполнялось следующее условие:

- Из любого города можно добраться до столицы, воспользовавшись телепортами ровно k раз.

Найдите минимальное количество телепортов, которые нужно перенаправить, чтобы описанное условие выполнялось.

Формат входных данных

Первая строка содержит два целых числа n и k ($2 \leqslant n \leqslant 100\,000$, $1 \leqslant k \leqslant 10^9$) — количество городов и любимое число короля.

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant n$) — номера городов, в которые направлены телепорты из каждого города королевства. Гарантируется, что, пользуясь телепортами, из любого города можно добраться до столицы.

Формат выходных данных

Выведите одно целое число — минимальное количество телепортов, которые нужно перенаправить, чтобы описанное условие выполнялось.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 1	2
4 2 1 1 2 2	0
8 2 4 1 2 3 1 2 3 4	3

Замечание

В первом примере достаточно направить все телепорты в столицу — город с номером 1.

Во втором примере из любого города можно добраться до столицы, воспользовавшись телепортом 2 раза. Поэтому можно не менять направления телепортов.

В третьем примере можно направить телепорты следующим образом: $a = (1, 1, 2, 1, 1, 2, 2, 4)$.

Задача B. Пути

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Дан простой связный неориентированный граф на n вершинах и m ребрах. Вершины пронумерованы целыми числами от 1 до n, а ребра — целыми числами от 1 до m, i-е ребро соединяет вершины u_i и v_i.

Также дано множество номеров ребер $S = \{x_1, x_2, \ldots, x_k\}$.

Необходимо определить, существует ли некоторый путь (не обязательно простой), который проходит по ребру x_i ровно один раз для каждого i от 1 до k.

Формат входных данных

Первая строка содержит два целых числа n и m ($2 \leqslant n \leqslant 200\,000$, $n - 1 \leqslant m \leqslant \min\left(\frac{n(n-1)}{2}, 200\,000\right)$) — количество вершин и ребер в графе.

Каждая из следующих m строк содержит два целых числа u_i и v_i ($1 \leqslant u_i < v_i \leqslant n$) — вершины, соединенные i-м ребром. Гарантируется, что все ребра попарно различны и граф является связным.

Следующая строка содержит одно целое число k ($1 \leqslant k \leqslant m$) — количество ребер в множестве S.

Последняя строка содержит k целых чисел $x_1, x_2, \ldots, x_k$ ($1 \leqslant x_1 < x_2 < \ldots < x_k \leqslant m$) — номера ребер, включенных в множество S.

Формат выходных данных

Выведите «`Yes`», если искомый путь существует, и «`No`» в противном случае.

Примеры

стандартный ввод	стандартный вывод
6 6 1 3 2 3 3 4 4 5 4 6 5 6 4 1 2 4 5	Yes
6 5 1 2 1 3 1 4 1 5 1 6 3 1 2 3	No

Замечание

В первом примере можно выбрать путь $1 \to 3 \to 4 \to 5 \to 6 \to 4 \to 3 \to 2$.

Задача C. Совершенное паросочетание в дереве

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Дано дерево T, состоящее из n вершин, пронумерованных числами от 1 до n. Дерево задано $n-1$ ребрами, i-е ребро соединяет вершины u_i и v_i. Ребра являются неориентированными.

Построим взвешенный полный граф G на n вершинах следующим образом:

- Вес ребра $w(x, y)$ между вершинами x и y в графе G равен расстоянию между вершинами x и y в дереве T.

Необходимо найти максимальное по весу полное паросочетание в графе G. Иными словами, необходимо найти $\lfloor \frac{n}{2} \rfloor$ пар вершин $M = \{(x_1, y_1), (x_2, y_2), \ldots, (x_{\lfloor \frac{n}{2} \rfloor}, y_{\lfloor \frac{n}{2} \rfloor})\}$, таких, что каждая из вершин от 1 до n появляется в M не более одного раза, и сумма $\sum\limits_{i=1}^{\lfloor \frac{n}{2} \rfloor} w(x_i, y_i)$ максимальна.

Формат входных данных

Первая строка содержит одно целое число n ($2 \leqslant n \leqslant 200\,000$) — количество вершин в дереве.

Каждая из следующих $n-1$ строк содержит два целых числа u_i и v_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$) — вершины, соединенные i-м ребром.

Формат выходных данных

Выведите $\lfloor \frac{n}{2} \rfloor$ строк, в i-й строке выведите два целых числа x_i и y_i ($1 \leqslant x_i, y_i \leqslant n$, $x_i \neq y_i$) — вершины, соединенные i-м ребром оптимального паросочетания.

Если оптимальных ответов несколько, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод

`4`
`1 2`
`2 3`
`3 4`
	`4 1`
	`3 2`

`3`
`1 2`
`2 3`
	`1 3`

Замечание

В первом примере $w(2, 4) = 2$, а $w(1, 3) = 2$. Таким образом, вес паросочетания $\{(2, 4), (1, 3)\}$ равен 4.

Во втором примере $w(1, 3) = 2$.

Задача D. Мосты

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Даны два массива $a_1, a_2, \ldots, a_m$ и $b_1, b_2, \ldots, b_m$, состоящие из целых чисел от 1 до n.

Рассмотрим некоторую строку длины m, состоящую из символов `0` и `1`. Построим по ней неориентированный граф на $2n$ вершинах и $n + m$ ребрах следующим образом:

- Если i-й символ строки равен `0`, i-е ребро соединяет вершины a_i и $b_i + n$;
- Если i-й символ строки равен `1`, i-е ребро соединяет вершины b_i и $a_i + n$;
- $(j + m)$-е ребро соединяет вершины j и $j + n$ для всех $1 \leqslant j \leqslant n$.

Ваша задача — найти такую строку длины m, состоящую из `0` и `1`, что в соответствующем ей графе будет минимальное количество мостов.

Формат входных данных

Первая строка содержит два целых числа n и m $(1 \leqslant n, m \leqslant 200\,000)$ — ограничение на числа в массивах, а также длина массивов.

Вторая строка содержит m целых чисел $a_1, a_2, \ldots, a_m$ $(1 \leqslant a_i \leqslant n)$ — первый массив.

Третья строка содержит m целых чисел $b_1, b_2, \ldots, b_m$ $(1 \leqslant b_i \leqslant n)$ — второй массив.

Формат выходных данных

Выведите искомую бинарную строку. Если оптимальных строк несколько, выведите любую из них.

Примеры

стандартный ввод	стандартный вывод
2 2 1 1 2 2	01
6 7 1 1 2 3 4 4 5 2 3 3 4 5 6 6	0100010

Замечание

Рассмотрим первый пример.

Граф, соответствующий строке `01`, не имеет мостов.

В графе, соответствующем строке `00`, ребра $(1, 3)$ и $(2, 4)$ являются мостами.

Задача E. Удаление камней

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

Дано дерево на n вершинах, пронумерованных числами от 1 до n. Дерево задано $n-1$ ребрами, i-е ребро соединяет вершины a_i и b_i.

Изначально в i-й вершине лежит s_i камней. Необходимо определить, можно ли удалить все камни из всех вершин, выполнив следующую операцию некоторое количество раз:

- Выбрать два различных листа в дереве, а затем удалить ровно один камень из каждой вершины на пути между выбранными листьями.

Листом называется вершина, степень которой равна 1. Обратите внимание, что операцию нельзя выполнить, если на пути между выбранными листьями есть вершина, в которой нет камней. Сами выбранные листья также считаются частью пути.

Формат входных данных

Первая строка содержит одно целое число n ($2 \leqslant n \leqslant 100\,000$) — количество вершин в дереве.

Вторая строка содержит n целых чисел $s_1, s_2, \ldots, s_n$ ($0 \leqslant s_i \leqslant 10^9$) — количество камней в каждой вершине.

Каждая из следующих $n-1$ строк содержит два целых числа a_i и b_i ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$) — вершины, соединенные i-м ребром.

Формат выходных данных

Выведите «`Yes`», если существует последовательность действий, после которой все камни будут удалены. В противном случае выведите «`No`».

Примеры

стандартный ввод	стандартный вывод
5 1 2 1 1 2 2 4 5 2 3 2 1 3	Yes
3 1 2 1 1 2 2 3	No
6 3 2 2 2 2 2 1 2 2 3 1 4 1 5 4 6	Yes

Замечание

В первом примере можно удалить все камни следующим образом:

1. Выбрать вершины 4 и 5. После этого в каждой вершине, кроме вершины 4, останется ровно 1 камень;
2. Выбрать вершины 1 и 5. После этого в дереве не останется камней.

Задача F. Черно-белое дерево

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

Есть дерево, состоящее из n вершин, пронумерованных числами от 1 до n. Вершины соединены $n - 1$ ребрами, i-е ребро соединяет вершины с номерами a_i и b_i.

Изначально все вершины не окрашены.

Алиса и Боб играют в игру, раскрашивая вершины. Они по очереди выполняют следующую операцию, начиная с Алисы:

- Выбрать вершину, которая еще не окрашена. Если ход делает Алиса, она красит вершину в белый цвет; если ход делает Боб, он красит ее в черный.

После того как все вершины будут окрашены, выполняется следующая процедура:

- Все белые вершины, соседние хотя бы с одной черной вершиной, перекрашиваются в черный цвет. Обратите внимание, что все такие белые вершины перекрашиваются одновременно, а не по одной.

Если после этого остается хотя бы одна белая вершина, побеждает Алиса. Если же все вершины становятся черными, побеждает Боб.

Определите, кто победит при оптимальной игре обоих игроков.

Формат входных данных

Первая строка содержит одно целое число n ($2 \leqslant n \leqslant 100\,000$) — количество вершин дерева.

Каждая из следующих $n - 1$ строк содержит два целых числа a_i и b_i ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$) — вершины, соединенные i-м ребром. Гарантируется, что данные ребра образуют дерево.

Формат выходных данных

Выведите «`First`», если при оптимальной игре победит Алиса, и «`Second`», если победит Боб.

Примеры

стандартный ввод	стандартный вывод
3 1 2 2 3	First
4 1 2 2 3 2 4	First
6 1 2 2 3 3 4 2 5 5 6	Second

Замечание

Рассмотрим один из вариантов развития событий для первого примера:

1. Алиса покрасит вершину 2 в белый цвет;

2. Боб покрасит вершину 1 в черный цвет;

3. Алиса покрасит вершину 3 в белый цвет.

В конце игры вершина 2 будет перекрашена в черный цвет, так как соседняя с ней вершина 1 окрашена в черный цвет. Однако вершина 3 останется белой, поэтому победит Алиса.

Задача G. Разноцветное дерево

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	1024 мегабайта

Дано дерево, состоящее из n вершин, пронумерованных числами от 1 до n. Дерево задается $n-1$ ребрами, i-е ребро соединяет вершины a_i и b_i. Цвет i-го ребра равен c_i, а его длина — d_i.

Необходимо ответить на q запросов:

- Предположим, что длина каждого ребра, цвет которого равен x_j, стала равна y_j. Необходимо найти расстояние между вершинами u_j и v_j.

Обратите внимание, что изменения длин ребер не сохраняются и влияют только на текущий запрос.

Формат входных данных

Первая строка содержит два целых числа n и q ($2 \leqslant n \leqslant 100\,000$, $1 \leqslant q \leqslant 100\,000$) — количество вершин в дереве и количество запросов.

Каждая из следующих $n-1$ строк содержит четыре целых числа a_i, b_i, c_i и d_i ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$, $1 \leqslant c_i \leqslant n-1$, $1 \leqslant d_i \leqslant 10\,000$) — вершины, соединенные i-м ребром, цвет i-го ребра и длину i-го ребра.

Каждая из следующих q строк содержит четыре целых числа x_i, y_i, u_i и v_i ($1 \leqslant x_i \leqslant n-1$, $1 \leqslant y_i \leqslant 10\,000$, $1 \leqslant u_i < v_i \leqslant n$) — параметры i-го запроса.

Формат выходных данных

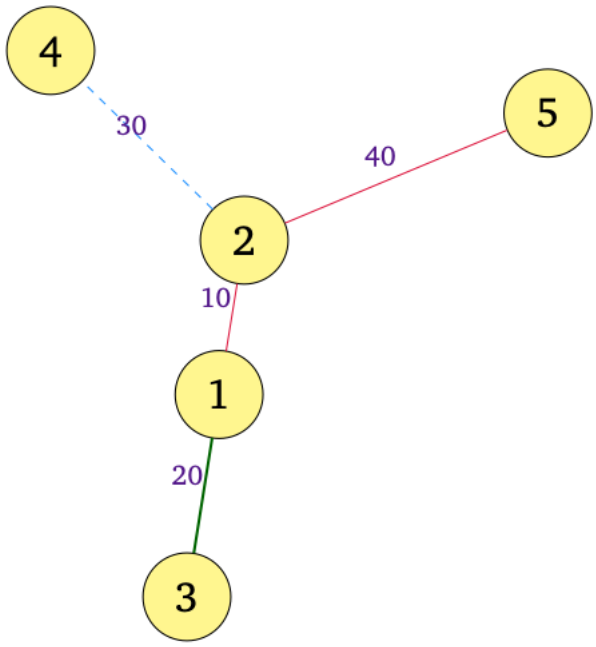
Выведите q целых чисел — ответы на запросы.

Пример

стандартный ввод	стандартный вывод
5 3 1 2 1 10 1 3 2 20 2 4 4 30 5 2 1 40 1 100 1 4 1 100 1 5 3 1000 3 4	130 200 60

Замечание

Дерево из примера выглядит следующим образом:



Ребра цвета 1 обозначены красными линиями, ребро цвета 2 обозначено зеленым цветом, а ребро цвета 4 — синим пунктиром.

Рассмотрим запросы:

1. Длина каждого ребра цвета 1 заменяется на 100. Расстояние между 1 и 4 равно $100+30 = 130$.
2. Длина каждого ребра цвета 1 заменяется на 100. Расстояние между 1 и 5 равно $100+100 = 200$.
3. Длина каждого ребра цвета 3 заменяется на $1\,000$. Расстояние между 3 и 4 равно $20 + 10 + 30 = 60$.

Задача H. k-цветные пути в дереве

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Дано дерево, состоящее из n вершин, пронумерованных числами от 1 до n. Дерево задано $n-1$ ребрами, i-е ребро соединяет вершины a_i и b_i. Вершины окрашены в цвета, цвет i-й вершины равен c_i.

Для каждого $k = 1, 2, \ldots, n$ необходимо решить следующую задачу:

- Найти количество простых путей, которые проходят хотя бы через одну вершину цвета k.

Обратите внимание, что простые пути из u в v и из v в u не считаются различными.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 200\,000$) — количество вершин в дереве.

Вторая строка содержит n целых чисел $c_1, c_2, \ldots, c_n$ ($1 \leqslant c_i \leqslant n$) — цвета вершин.

Каждая из следующих $n-1$ строк содержит два целых числа a_i и b_i ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$) — вершины, соединенные i-м ребром.

Формат выходных данных

Выведите n целых чисел: ответы на задачу при $k = 1, 2, \ldots, n$.

Примеры

стандартный ввод	стандартный вывод
3 1 2 1 1 2 2 3	5 4 0
1 1	1
2 1 2 1 2	2 2
5 1 2 3 4 5 1 2 2 3 3 4 3 5	5 8 10 5 5
8 2 7 2 5 4 1 7 5 3 1 1 2 2 7 4 5 5 6 6 8 7 8	18 15 0 14 23 0 23 0

Замечание

Рассмотрим первый пример.

Пути, проходящие через вершину, окрашенную в цвет 1: $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 3)$, $(3, 3)$.
Пути, проходящие через вершину, окрашенную в цвет 2: $(1, 2)$, $(1, 3)$, $(2, 2)$, $(2, 3)$.
Путей, проходящих через вершину, окрашенную в цвет 3, в дереве нет.