

Задача А. LCA - 2

Имя входного файла: lca2.in
Имя выходного файла: lca2.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Задано подвешенное дерево, содержащее n вершин, пронумерованных от 0 до $n - 1$. Требуется ответить на m запросов о наименьшем общем предке для пары вершин.

Запросы генерируются следующим образом. Заданы числа a_1, a_2 и числа x, y и z .

Числа a_3, \dots, a_{2m} генерируются следующим образом: $a_i = (x \cdot a_{i-2} + y \cdot a_{i-1} + z) \bmod n$. Первый запрос имеет вид $\langle a_1, a_2 \rangle$. Если ответ на $i - 1$ -й запрос равен v , то i -й запрос имеет вид $\langle (a_{2i-1} + v) \bmod n, a_{2i} \rangle$.

Формат входных данных

Первая строка содержит два числа: n ($1 \leq n \leq 100\,000$) и m ($1 \leq m \leq 10\,000\,000$). Корень дерева имеет номер 0. Вторая строка содержит $n - 1$ целых чисел, i -е из этих чисел это предок вершины i

Третья строка содержит целые числа a_1 и a_2 ($0 \leq a_i \leq n - 1$).

Четвёртая строка содержит три целых числа: x, y и z ($0 \leq x, y, z \leq 10^9$)

Формат выходных данных

Выведите в выходной файл сумму номеров вершин — ответов на все запросы.

Примеры

lca2.in	lca2.out
3 2 0 1 2 1 1 1 0	2
1 2 0 0 1 1 1	0

Задача В. Учиться!

Имя входного файла: `moscow.in`
Имя выходного файла: `moscow.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Каждый год огромное количество выпускников, сдавшие ЕГЭ, выбирают, куда же они пойдут учиться. Не удивительно, что многие из них предпочитают перебраться поближе к столице. Транспортная инфраструктура страны переживает не лучшие времена, и в приемлемом качестве поддерживается минимально возможное число городов, необходимое для того, чтобы от любого города можно было добраться до любого другого.

Каждый выпускник оценивает свои результаты сдачи экзаменов, и решает, насколько далеко от своего родного города в сторону столицы он сможет уехать.

Выпускников настолько много, что вам не требуется выводить для каждого из них, до какого города он сможет доехать. Достаточно вывести сумму ответов для каждого выпускника.

Запросы генерируются следующим образом. Заданы числа a_1, a_2 и числа x, y и z . Числа a_3, \dots, a_{2m} генерируются следующим образом: $a_i = (x \cdot a_{i-2} + y \cdot a_{i-1} + z) \bmod n$. Первый запрос имеет вид $\langle a_1, a_2 \rangle$. Если ответ на $i-1$ -й запрос равен v , то i -й запрос имеет вид $\langle (a_{2i-1} + v) \bmod n, a_{2i} \rangle$. В i -м запросе первое число соответствует городу, в котором окончил школу i -й выпускник, а второе — насколько далеко от родного города он может уехать. Все выпускники стараются перебраться как можно ближе к столице.

Формат входных данных

Первая строка содержит два числа: n ($1 \leq n \leq 100\,000$) и m ($1 \leq m \leq 10\,000\,000$). Столица имеет номер 0. Вторая строка содержит $n-1$ целых чисел, i -е из этих чисел равно номеру следующего за городом i на пути к столице. Третья строка содержит два целых числа в диапазоне от 0 до $n-1$: a_1 и a_2 . Четвертая строка содержит три целых числа: x, y и z , эти числа неотрицательны и не превосходят 10^9 .

Формат выходных данных

Выведите в выходной файл сумму номеров городов — ответов на все запросы.

Примеры

<code>moscow.in</code>	<code>moscow.out</code>
3 2 0 1 2 1 1 1 0	1
1 2 0 0 1 1 1	0

Задача С. Динамический Лес

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

Вам нужно научиться обрабатывать 3 типа запросов:

1. Добавить ребро в граф (`link`).
2. Удалить ребро из графа (`cut`).
3. По двум вершинам a и b , определить, лежат ли они в одной компоненте связности (`get`).

Изначально граф пустой (содержит N вершин, не содержит ребер). Гарантируется, что в любой момент времени граф является лесом. При добавлении ребра гарантируется, что его сейчас в графе нет. При удалении ребра гарантируется, что оно уже добавлено.

Формат входных данных

Числа N и M ($1 \leq N \leq 10^5 + 1$, $1 \leq M \leq 10^5$) — количество вершин в дереве и, соответственно, запросов. Далее M строк, в каждой строке команда (`link` или `cut`, или `get`) и 2 числа от 1 до N — номера вершин в запросе.

Формат выходных данных

В выходной файл для каждого запроса `get` выведите 0, если не лежат, или 1, если лежат.

Примеры

стандартный ввод	стандартный вывод
3 7 get 1 2 link 1 2 get 1 2 cut 1 2 get 1 2 link 1 2 get 1 2	0101
5 10 link 1 2 link 2 3 link 4 3 cut 3 4 get 1 2 get 1 3 get 1 4 get 2 3 get 2 4 get 3 4	110100

Задача D. k -е ребро в поддереве

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

Рассмотрим дерево из n вершин, пронумерованных числами от 1 до n . На каждом ребре написано число от 1 до 10^9 .

Вам нужно ответить на следующие запросы: для некоторого поддерева, выпишите значения, написанные на ребрах и найдите k -е среди них. Каждое поддерево задается подмножеством из z вершин. Возьмем объединение кратчайших путей между всеми парами этих вершин. Иначе говоря, ребро принадлежит поддереву тогда и только тогда, когда оно лежит на хотя бы одном кратчайшем пути между парой заданных вершин.

Формат входных данных

Первая строка содержит число n , количество вершин в дереве ($2 \leq n \leq 10^5$).

Следующие $n - 1$ строк содержат по три числа каждая; i -я из них содержит числа a_i , b_i и c_i : номера вершины, соединенных i -м ребром, и число, которое написано на этом ребре ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$, $1 \leq c_i \leq 10^9$).

Следующая строка содержит число q : количество запросов ($1 \leq q \leq 10^5$).

Каждая из следующих q строк описывает очередной запрос. Множество из z вершин задается z различными числами v_1, v_2, \dots, v_z , после которых записан 0 ($1 \leq z \leq n$, $1 \leq v_j \leq n$). После этого записано число k ($1 \leq k \leq n - 1$). Смотрите примеры в условии для пояснения.

Сумма z по всем запросам не превосходит 10^5 .

Формат выходных данных

Для каждого запроса выведите ответ в отдельной строке: значение k -го ребра в заданном поддереве. Если в поддереве меньше, чем k ребер, выведите -1 .

Примеры

стандартный ввод	стандартный вывод
6	2
1 2 4	3
2 3 2	1
3 4 2	-1
3 5 1	-1
6 5 3	
5	
1 2 5 6 0 2	
2 4 6 0 4	
6 4 0 1	
1 3 0 3	
3 0 5	

Задача Е. 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: $f(2), f(3), \dots, f(n)$.

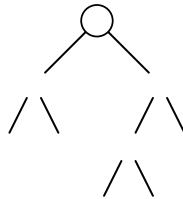
Примеры

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

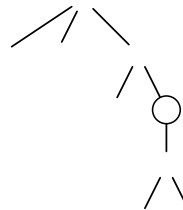
Задача F. Dynamic LCA

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

Постановка задачи о *наименьшем общем предке* прежде такова: дано дерево T с выделенным корнем и две вершины u и v , $\text{lca}(u, v)$ — вершина с максимальной глубиной, которая является предком и u , и v . Например, на картинке внизу $\text{lca}(8, 7)$ — вершина 3.



С помощью операции $\text{chroot}(u)$ мы можем менять корень дерева, достаточно отметить u , как новый корень, и направить ребра вдоль пути от корня. Наименьшие общие предки вершин поменяются соответственно. Например, если мы сделаем $\text{chroot}(6)$ на картинке сверху, $\text{lca}(8, 7)$ станет вершина 6. Получившееся дерево изображено внизу.



Вам дано дерево T . Изначально корень этого дерева — вершина 1. Напишите программу, которая поддерживает эти две операции: $\text{lca}(u, v)$ и $\text{chroot}(u)$.

Формат входных данных

Входной файл состоит из нескольких тестов.

Первая строка каждого теста содержит натуральное число n — количество вершин в дереве ($1 \leq n \leq 100\,000$). Следующие $n - 1$ строк содержат по 2 натуральных числа и описывают ребра дерева. Далее идет строка с единственным натуральным числом m — число операций. Следующие m строк содержат операции. Строка $? u v$ означает операцию $\text{lca}(u, v)$, а строка $! u$ — $\text{chroot}(u)$. Последняя строка содержит число 0.

Сумма n для всех тестов не превосходит 100 000. Сумма m для всех тестов не превосходит 200 000.

Формат выходных данных

Для каждой операции $? u v$ выведите значение $\text{lca}(u, v)$. Числа разделяйте переводами строк.

Примеры

dynamic.in	dynamic.out
9	2
1 2	1
1 3	3
2 4	6
2 5	2
3 6	3
3 7	6
6 8	2
6 9	
10	
? 4 5	
? 5 6	
? 8 7	
! 6	
? 8 7	
? 4 5	
? 4 7	
? 5 9	
! 2	
? 4 3	
0	

Задача G. Divljak

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	768 мегабайт

Это история про N варваров и Тарзана. Варвары занумерованы числами от 1 до N , и у каждого на табличке написана строка из строчных букв латинского алфавита. Игра состоит из Q раундов, каждый из которых может быть одного из двух типов:

1. Тарзан показывает слово P варварам.
2. Тарзан спрашивает варвара номер i , сколько слов из показанных им до текущего момента таковы, что слово на табличке варвара является их подстрокой.

Варвары не очень любят задачи на строчки, поэтому помогите им ответить на все запросы Тарзана.

Формат входных данных

Первая строка содержит число варваров N ($1 \leq N \leq 10^5$).

Каждая из следующих N строк содержит слово на табличке очередного варвара.

Далее идет число запросов Q ($1 \leq Q \leq 10^5$).

Каждая из следующих Q строк описывает очередной запрос. Сперва записано число t — тип запроса. Если t равно 1, то далее записано слово P , которое показывает Тарзан. Иначе t равно 2 и далее записан номер варвара i ($1 \leq i \leq N$), которого спрашивает Тарзан.

Суммарная длина слов на табличках варваров $\leq 2 \cdot 10^6$.

Суммарная длина показанных Тарзаном слов $\leq 2 \cdot 10^6$.

Формат выходных данных

Для каждого раунда второго типа выведите ответ на вопрос Тарзана.

Примеры

стандартный ввод	стандартный вывод
3 a bc abc 3 1 abca 2 1 2 3	1 1
7 abba bbaa b bbaa abba a ba 7 1 aaabbabbaab 2 7 1 baabaaa 1 aabbbab 2 3 1 aabba 2 3	1 3 4