

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. Самое дешёвое ребро

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

Дано подвешенное дерево с корнем в первой вершине. Все ребра имеют веса (стоимости). Вам нужно ответить на m запросов вида «найти ребро минимального веса на пути между двумя заданными вершинами».

Формат входных данных

В первой строке файла записано одно число n — количество вершин в дереве ($1 \leq n \leq 50000$).

В i -й из следующих $n-1$ строк записано два целых числа p_{i+1} и w_{i+1} ($p_{i+1} < i+1$, $|w_{i+1}| \leq 10^6$) — предок вершины $i+1$ и вес ребра из вершины $i+1$ в её предка.

Следующая строка содержит число m ($0 \leq m \leq 5 \cdot 10^4$) — количество запросов. Следующие m строк содержат по одному запросу вида (x_i, y_i) — найти минимальный вес ребра на пути из вершины x_i в вершину y_i ($1 \leq x_i \leq n$, $1 \leq y_i \leq n$, $x_i \neq y_i$).

Формат выходных данных

Программа должна вывести m чисел — ответы на запросы.

Примеры

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

Задача С. LCA-3

Имя входного файла: lca3.in
Имя выходного файла: lca3.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Подвешенное дерево — это ориентированный граф без циклов, в котором в каждую вершину, кроме одной, называемой *корнем* ориентированного дерева, входит одно ребро. В корень ориентированного дерева не входит ни одного ребра. *Отцом* вершины называется вершина, ребро из которой входит в данную.

(по материалам Wikipedia)

Дан набор подвешенных деревьев. Требуется выполнять следующие операции:

- 0 u v Для двух заданных вершин u и v выяснить, лежат ли они в одном дереве. Если это так, вывести вершину, являющуюся их наименьшим общим предком, иначе вывести 0.
- 1 u v Для корня u одного из деревьев и произвольной вершины v другого дерева добавить ребро (v, u) . В результате эти два дерева соединятся в одно.

Вам необходимо выполнять все операции online, т.е. вы сможете узнать следующий запрос только выполнив предыдущий.

Формат входных данных

На первой строке входного файла находится число n — суммарное количество вершин в рассматриваемых деревьях, $1 \leq n \leq 50000$. На следующей строке расположено n чисел — предок каждой вершины в начальной конфигурации, или 0, если соответствующая вершина является корнем. Затем следует число k — количество запросов к вашей программе, $1 \leq k \leq 100000$. Каждая из следующих строк содержит по три целых числа: вид запроса (0 — для поиска LCA или 1 — для добавления ребра) и два числа x, y . Вершины, участвующие в запросе можно вычислить по формуле: $u = (x - 1 + ans) \bmod n + 1$, $v = (y - 1 + ans) \bmod n + 1$, где ans - ответ на последний запрос типа 0 ($ans = 0$ для первого запроса).

Формат выходных данных

Для каждого запроса типа 0, выведите в выходной файл одно число на отдельной строке — ответ за этот запрос.

Примеры

lca3.in	lca3.out
5	0
0 0 0 0 0	5
12	5
1 5 3	3
0 2 5	2
1 4 2	3
1 1 5	3
0 1 5	2
1 3 4	
0 1 5	
0 3 1	
0 4 2	
0 1 4	
0 5 2	
0 4 1	

Задача D. LCA

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

Дан ориентированный граф. Подсчитайте, сколько пар вершин (i, j) имеют общего предка. Общим предком вершин i и j называется такая вершину k , что и i , и j достижимы из k .

Формат входных данных

В первой строке входного файла содержатся целые числа $1 \leq N \leq 10^4, 0 \leq M \leq 10^4$ — количество вершин и рёбер в графе. Далее следуют M строк по два числа от 1 до N . Пара чисел (a, b) означает, что из вершины a есть ребро в вершину b .

Формат выходных данных

Выведите одно целое число — количество пар.

Примеры

<code>lca.in</code>	<code>lca.out</code>
2 1 1 2	4
3 2 2 1 3 1	7

Задача Е. 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 F. Максимумы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 10 секунд
Ограничение по памяти: 512 мегабайт

К сожалению, у Дани не хватило времени, чтобы написать нескучное условие к этой задаче.

Вам дан массив a_0, \dots, a_{n-1} , который задаётся числами n, a_0, b, c, d следующим образом:

$$b_i = (b_{i-1} \cdot c + d) \bmod 2^{31} \text{ для } i \geq 1$$

$$a_i = (a_{i-1} + 1 - 2 \cdot ((b_i \bmod 239179) \bmod 2)) \text{ для } i \geq 1$$

Обратите внимание, что два соседних числа отличаются либо на $+1$, либо на -1 .

Ответьте на n запросов, i -й запрос — максимум на отрезке $[\min(l_i, r_i), \max(l_i, r_i)]$ для $i = 0 \dots n-1$.

Пусть ans_i — ответ на i -й запрос. Будем считать, что $ans_{-1} = 0$. Вам задаётся число x_0 . Далее, l_i и r_i, x_i вычисляются так:

$$l_i = (x_i + ans_{i-1}) \bmod n \text{ для } i \geq 0$$

$$r_i = (l_i + i) \bmod n \text{ для } i \geq 0$$

$$x_i = (x_{i-1} \cdot 1103515245 + 12345) \bmod 2^{31} \text{ для } i \geq 1$$

При взятии по модулю обратите внимание на то, что ans_i бывают отрицательными.

Формат входных данных

Единственная строка ввода содержит шесть целых чисел n, a_0, b, c, d, x_0 .

- $1 \leq n \leq 2 \cdot 10^7$
- $-10^9 \leq a_0 \leq 10^9$
- $0 \leq b, c, d, x_0 \leq 2^{31} - 1$

Формат выходных данных

Выведите сумму ответов на все запросы.

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4 0	11
100500 -1 23 45 67 89	-8614564

Задача G. Дерево

Имя входного файла: `tree.in`
Имя выходного файла: `tree.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Задано подвешенное дерево, содержащее n ($1 \leq n \leq 1\,000\,000$) вершин. Каждая вершина покрашена в один из n цветов. Требуется для каждой вершины v вычислить количество различных цветов, встречающихся в поддереве с корнем v .

Формат входных данных

В первой строке входного файла задано число n .

Следующие n строк описывают вершины, по одной в строке. Описание очередной вершины i имеет вид $p_i c_i$, где p_i — номер родителя вершины i , а c_i — цвет вершины i ($1 \leq c_i \leq n$). Для корня дерева $p_i = 0$.

Формат выходных данных

Выведите n чисел, обозначающих количества различных цветов в поддеревьях с корнями в вершинах $1, \dots, n$.

Примеры

tree.in	tree.out
5	1 2 3 1 1
2 1	
3 2	
0 3	
3 3	
2 1	

Задача Н. Кварум

Имя входного файла:	stdin
Имя выходного файла:	stdout
Ограничение по времени:	7.5 секунды
Ограничение по памяти:	512 мегабайт

Второе августа 1702 года, среда. На болоте, ныне называемом Санкт-Петербург, собрались лягушки со всех ближайших водоемов. Для каждой из жабок Великим Уквакнителем была приготовлена определенная кувшинка. Оказалось, что лягушки могут общаться только с теми лягушками, кто сидит на лягушке похожего цвета. Давайте соединим такие кувшинки ребрами. Оказалось, что полученный граф является подвешенным деревом на n вершинах (в корне дерева сидит сам Великий Уквакнитель). Все кувшинки пронумерованы от 0 до $n - 1$.

Так как лягушки собрались, чтобы обсудить разрешение людям на строительство на их любимом болоте города, то некоторые пары лягушек хотят обмениваться своими аргументами друг с другом. Для ускорения процесса они просят вас найти их наименьшего общего предка.

Номера лягушек, которые в момент времени k ($1 \leq k \leq m$) хотят получить от вас их наименьшего предка, это a_{2k-1} и a_{2k} генерируются следующим образом:

- Изначально даны a_1, a_2 и числа x, y, 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 $\langle a_1, a_2 \rangle$. i -й запрос имеет вид $\langle a_{2i-1}, a_{2i} \rangle$.

Формат входных данных

Первая строка содержит два числа: n ($1 \leq n \leq 10^6$) и m ($1 \leq m \leq 10^7$).

Великий Уквакнитель сидит на кувшинке с номером 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

stdin	stdout
3 2 0 1 2 1 1 1 0	1
1 2 0 0 1 1 1	0

Замечание

Сюда надо сдать Тарьяна. Остальные решения получают реджект

Задача I. LCA

Имя входного файла: lca2.in
Имя выходного файла: lca2.out
Ограничение по времени: 6 секунд
Ограничение по памяти: 256 мегабайт

Дано подвешенное дерево с корнем в первой вершине. Вам нужно ответить на m запросов вида “найти LCA двух вершин”.

LCA вершин u и v в подвешенном дереве — это наиболее удалённая от корня дерева вершина, лежащая на обоих путях от u и v до корня.

Формат входных данных

В первой строке задано целое число n — число вершин в дереве ($1 \leq n \leq 2 \cdot 10^5$).

В следующих $n - 1$ строках записано одно целое число x . Число x на строке i означает, что x — предок вершины i ($x < i$).

Затем дано число m .

Далее заданы m ($0 \leq m \leq 5 \cdot 10^5$) запросов вида (u, v) — найти LCA двух вершин u и v ($1 \leq u, v \leq n; u \neq v$).

Формат выходных данных

Для каждого запроса выведите LCA двух вершин на отдельной строке.

Примеры

lca2.in	lca2.out
5	1
1	1
1	
2	
3	
2	
2 3	
4 5	
5	2
1	2
1	1
2	
2	
3	
4 5	
4 2	
3 5	