Задача 1.

Лёгкая

**Ancestor. Предок**

Напишите программу, которая для двух вершин дерева определяет, является ли одна из  
них предком другой.

ВХОДНЫЕ ДАННЫЕ

Первая строка входного файла содержит натуральное число n (1≤n≤100000)~---  
количество вершин в дереве.  
Во второй строке находится n чисел. При этом i-ое число второй строки  
определяет непосредственного родителя вершины с номером i. Если номер родителя  
равен нулю, то вершина является корнем дерева. В третьей строке находится число m (1≤m≤100000) ~-— количество запросов.  
Каждая из следующих m строк содержит два различных числа a и b (1≤a,b≤n).

ВЫХОДНЫЕ ДАННЫЕ

Для каждого из m запросов выведите на отдельной строке число 1, если вершина a  
является одним из предков вершины b, и 0 в противном случае.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 6 0 1 1 2 3 3 5 4 1 1 4 3 6 2 6 6 5 | 0 1 1 0 0 |

Задача 2.

Сложная

**LCA**

Задано подвешенное дерево, содержащее n (1⩽n⩽100000)  
вершин, пронумерованных от 0 до n−1.  
Требуется ответить на m (1⩽m⩽10000000) запросов  
о наименьшем общем предке для пары вершин. Запросы генерируются следующим образом. Заданы числа  
a1,a2 и числа x, y и z.  
Числа a3,…,a2m генерируются  
следующим образом: ai=(x⋅ai−2+y⋅ai−1+z)modn.  
Первый запрос имеет вид ⟨a1,a2⟩. Если ответ на i−1-й запрос  
равен v, то i-й запрос имеет вид  
⟨(a2i−1+v)mod n,a2i⟩.

ВХОДНЫЕ ДАННЫЕ

Первая строка содержит два числа: n и m. Корень дерева имеет номер 0. Вторая строка содержит n−1 целых чисел, i-е из этих  
чисел равно номеру родителя вершины i.  
Третья строка содержит два целых числа в диапазоне от 0 до n−1: a1 и a2.  
Четвертая строка содержит три целых числа: x, y и z, эти числа неотрицательны  
и не превосходят 109.

ВЫХОДНЫЕ ДАННЫЕ

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

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 3 2 0 1 2 1 1 1 0 | 2 |

Задача 3.

Сложная

**Самое дешёвое ребро**

Дано подвешенное дерево с корнем в первой вершине. Все ребра имеют веса (стоимости).  
Вам нужно ответить на M запросов вида ``найти у двух вершин минимум среди стоимостей ребер пути между ними''.

ВХОДНЫЕ ДАННЫЕ

В первой строке файла записано количество вершин n (2⩽n⩽50000).  
В следующих n−1 строках записаны отцы вершин 2..n.  
Для вершины i записана пара x y. Число x означает, что x -— предок вершины i,y означает стоимость ребра.  
x<i,|y|⩽106. Далее число запросов m (0⩽m⩽50000) и m запросов вида (x,y) -— найти минимум на пути из x в y (x≠y).

ВЫХОДНЫЕ ДАННЫЕ

m ответов на запросы.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5 1 2 1 3 2 5 3 2 2 2 3 4 5 | 2 2 |

адача 4.

Сложная

**Цветные волшебники**

Сказочная страна представляет собой множество городов, соединенных дорогами с двухсторонним движением. Причем из любого города страны можно добраться в любой другой город либо непосредственно, либо через другие города. Известно, что в сказочной стране не существует дорог, соединяющих город сам с собой и между любыми двумя разными городами, существует не более одной дороги. В сказочной стране живут желтый и синий волшебники. Желтый волшебник, пройдя по дороге,  
перекрашивает ее в желтый цвет, синий -— в синий. Как известно, при наложении желтой краски на  
синюю, либо синей краски на желтую, краски смешиваются и превращаются в краску зеленого цвета, который является самым нелюбимым цветом обоих волшебников. В этом году в столице страны (городе f) проводится конференция волшебников. Поэтому желтый и синий волшебники хотят узнать, какое минимальное количество дорог им придется перекрасить в зеленый цвет, чтобы добраться в столицу. Изначально все дороги не покрашены. Начальное положение желтого и синего волшебников заранее не известно. Поэтому необходимо решить данную задачу для k возможных случаев их начальных расположений.

ВХОДНЫЕ ДАННЫЕ

Первая строка входного файла содержит целые числа: n (1⩽n⩽100000) и m  
(1⩽m⩽500000) -— количество городов и дорог в волшебной стране соответственно. Третья  
строка содержит одно целое число f (1⩽f⩽n) -— номер города, являющегося столицей сказочной страны. В следующих m строках, находится описание дорог страны. В этих m строк записано по два целых числа ai и bi, означающих, что существует дорога, соединяющая города ai и bi. Следующая строка содержит целое число k (1⩽k⩽100000) -— количество возможных начальных расположений волшебников. Далее следуют k строк, каждая из которых содержит два целых числа -— номера городов, в которых изначально  
находится желтый и синий волшебники соответственно.

ВЫХОДНЫЕ ДАННЫЕ

Для каждого из k случаев, ваша программа должна вывести в выходной минимальное количество дорог, которое придется покрасить в зеленый цвет волшебникам для того, чтобы добраться в столицу.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 6 6 1 1 2 2 3 3 4 4 2 4 5 3 6 2 5 6 6 6 | 1 2 |

Задача 5.

Олимпиадная

**MeX на пути дерева**

Вам дано дерево из n вершин. Каждому ребру дерева сопоставлено неотрицательное целое число xi.  
Ваша задача — ответить на q запросов. j-й из запросов — найти  
минимальное неотрицательное целое число y, отсутствующее на пути между вершинами aj и bj.

ВХОДНЫЕ ДАННЫЕ

Первая строка содержит числа n и q (2≤n≤105, 1≤q≤105), количество вершин в дереве и число запросов. Следующие n−1 строка содержит описания рёбер дерева, тройки целых чисел ui, vi, xi (1≤ui,vi≤n, ui≠vi, 0≤xi≤109). Следующие q строк содержат пары целых чисел aj, bj (1≤aj,bj≤n), обозначающие запросы.

ВЫХОДНЫЕ ДАННЫЕ

На каждый запрос aj, bj на отдельной строке выведите минимальное целое неотрицательное y, не лежащее на пути между aj и bj.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 7 6 2 1 1 3 1 2 1 4 0 4 5 1 5 6 3 5 7 4 1 3 4 1 2 4 2 5 3 5 3 7 | 0 1 2 2 3 3 |