

Задача А. Расстояние в дереве

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

Вам дано неориентированное дерево (связный граф без циклов) на N вершинах. Требуется отвечать на запросы вида «найти длину кратчайшего пути между данной парой вершин».

Формат входных данных

В первой строке находится одно целое число N ($1 \leq N \leq 10^5$) — количество вершин в дереве.

В каждой из следующих $N - 1$ строк задано по два числа, разделенных пробелом, u_i, v_i — вершины, соединенные i -ым ребром дерева ($1 \leq u_i, v_i \leq N$).

В следующей строке задано число M — количество запросов ($0 \leq M \leq 10^5$).

В каждой из следующих M строк задано по два числа — номера вершин, соответствующих очередному запросу.

Формат выходных данных

Выведите M строк, в каждой из которых содержится одно целое число — ответ на очередной запрос. Ответы необходимо вывести в порядке, соответствующем порядку запросов на входе программы.

Примеры

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

Задача В. Самое дешевое 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 у двух вершин минимум среди стоимостей ребер пути между ними”.

Формат входных данных

В первой строке файла записано одно число — n (количество вершин).

В следующих $n - 1$ строках записаны два числа — x и y . Число x на строке i означает, что x — предок вершины i , y означает стоимость ребра.

$x < i$, $|y| \leq 10^6$.

Далее m запросов вида (x, y) — найти минимум на пути из x в y ($x \neq y$).

Ограничения: $2 \leq n \leq 5 \cdot 10^4$, $0 \leq m \leq 5 \cdot 10^4$.

Формат выходных данных

Выведите m ответов 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 - 2

Имя входного файла: lca2.in
Имя выходного файла: lca2.out
Ограничение по времени: 5 секунды
Ограничение по памяти: 256 мегабайт

Задано подвешенное дерево, содержащее n ($1 \leq n \leq 100\,000$) вершин, пронумерованных от 0 до $n-1$. Требуется ответить на m ($1 \leq m \leq 10\,000\,000$) 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 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 $n-1$: 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 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 D. ДНК Роботов

Имя входного файла: `robots.in`
Имя выходного файла: `robots.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Последние достижения в технологии синтеза ДНК позволили провести эксперимент по созданию биороботов.

Для облегчения задачи создания ПО для управления роботами было принято решение, что их ДНК будет состоять из $M = 2^n$ символов для некоторого $n \geq 2$. Кроме этого, по техническим причинам это будет не обычная строка, а циклическая, то есть её можно начинать читать с любой позиции.

Одной из целей эксперимента является изучение мутаций биороботов. В результате продолжительных наблюдений было найдено много различных видов роботов. Для понимания процесса мутации учёным необходимо решить следующую задачу. Для ДНК двух роботов требуется определить коэффициент их похожести. Он вычисляется, как максимальное количество совпадающих символов при наилучшем совмещении этих ДНК. Чем больше символов совпадает, тем лучше совмещение.

Требуется написать программу, которая найдёт наилучшее совмещение двух ДНК.

Формат входных данных

В первой строке входного файла записано одно число $M (4 \leq M \leq 131072)$. В следующих двух строках записаны ДНК двух роботов. Обе ДНК — строки, состоящие ровно из M символов из множества {'A', 'C', 'G', 'T'}.

Формат выходных данных

В выходной файл выведите два числа — максимальное количество совпадающих символов и значение оптимального сдвига — неотрицательное количество символов второй ДНК, которые необходимо перенести из конца строки в её начало для достижения наилучшего совмещения.

Примеры

<code>robots.in</code>	<code>robots.out</code>
16 ACGTACGTACGTACGT CGTACGTACGTACGTC	15 1