

Содержание

Задачи здорового человека	2
Задача 17A. Ancestor. Предок [0.3 sec, 256 mb]	2
Задача 17B. Дерево [2 sec, 256 mb]	3
Для искателей острых ощущений	4
Задача 17C. Самое дешевое ребро [1 sec, 256 mb]	4
Задача 17D. LCA Problem Revisited [2 sec, 256 mb]	5

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на GNU C++ компиляторы с суффиксом inc, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

Задачи здорового человека

Задача 17А. Ancestor. Предок [0.3 сек, 256 mb]

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

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 100\,000$) — 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 ($1 \leq m \leq 100\,000$) — количество запросов. Каждая из следующих m строк содержит два различных числа a и b ($1 \leq a, b \leq n$).

Формат выходных данных

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.

Пример

stdin	stdout
6	0
0 1 1 2 3 3	1
5	1
4 1	0
1 4	0
3 6	
2 6	
6 5	

Подсказка по решению

Мы умеем очень просто отвечать на такие запросы за $\mathcal{O}(1)$.

Задача 17В. Дерево [2 sec, 256 mb]

Дано взвешенное дерево. Найти кратчайшее расстояние между заданными вершинами.

Формат входных данных

Первая строка входного файла содержит натуральное число $N \leq 150\,000$ — количество вершин в графе. Вершины нумеруются целыми числами от 0 до $N - 1$. В следующих $N - 1$ строках содержится по три числа u, v, w , которые соответствуют ребру весом w , соединяющему вершины u и v . Веса — целые числа от 0 до 10^9 . В следующей строке содержится натуральное число $M \leq 75\,000$ — количество запросов. В следующих M строках содержится по два числа u, v — номера вершин, расстояние между которыми необходимо вычислить.

Формат выходных данных

Для каждого запроса выведите на отдельной строке одно число — искомое расстояние. Гарантируется, что ответ помещается в знаковом 32-битном целом типе.

Пример

stdin	stdout
3	0
1 0 1	1
2 0 1	1
9	1
0 0	0
0 1	2
0 2	1
1 0	2
1 1	0
1 2	
2 0	
2 1	
2 2	

Подсказка по решению

Вы же уже сдали LCA, да? Воспользуйтесь им. И частичными суммами.

Для искателей острых ощущений

Задача 17С. Самое дешевое ребро [1 сек, 256 mb]

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 числ — 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

m ответов на запросы.

Пример

stdin	stdout
5	2
1 2	2
1 3	
2 5	
3 2	
2	
2 3	
4 5	

Замечание

Двоичные подьёмы. Разбиралось на практике.

Задача 17D. LCA Problem Revisited [2 sec, 256 mb]

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

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

Подсказка по решению

Эйлеров обход и спарсы.