

Содержание

Must have	2
Задача 15A. Ancestor. Предок [0.2 sec, 256 mb]	2
Задача 15B. Общий предок [0.5 sec, 256 mb]	3
Обязательные задачи	4
Задача 15C. Дерево [1.2 sec, 256 mb]	4
Задача 15D. Самое дешевое ребро [0.5 sec, 256 mb]	5
Задача 15E. Dynamic LCA [1.5 sec, 256 mb]	6
Задача 15F. LCA Problem Revisited [1.6 sec, 256 mb]	7
Задача 15G. Перестановки [1.2 sec, 256 mb]	8
Дополнительные задачи	9
Задача 15H. Черепахи и повороты [1 sec, 256 mb]	9
Задача 15I. LCA-3 [1 sec, 256 mb]	10
Задача 15J. Дерево [2 sec, 256 mb]	11
Задача 15K. Фыр-Фыр-Фенвик [2 sec, 256 mb]	12

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь **примерами**.

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 компилятор GNU C++11 5.1.0 (TDM-GCC-64) inc, который позволяет пользоваться **дополнительной библиотекой**. Под ним можно сдать **вот это**.

Must have

Задача 15А. Ancestor. Предок [0.2 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	

Задача 15В. Общий предок [0.5 сек, 256 mb]

Дано подвешенное дерево с корнем в 1-й вершине и M запросов вида “найти у двух вершин наименьшего общего предка”.

Формат входных данных

В первой строке файла записано одно число N — количество вершин. В следующих $N - 1$ строках записаны числа. Число x на строке $2 \leq i \leq N$ означает, что x — отец вершин i . ($x < i$). На следующей строке число M . Следующие M строк содержат запросы вида (x, y) — найти наименьшего предка вершин x и y . Ограничения: $1 \leq N \leq 5 \cdot 10^4, 0 \leq M \leq 5 \cdot 10^4$.

Формат выходных данных

M ответов на запросы.

Пример

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

Обязательные задачи

Задача 15С. Дерево [1.2 сек, 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	

Задача 15D. Самое дешевое ребро [0.5 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

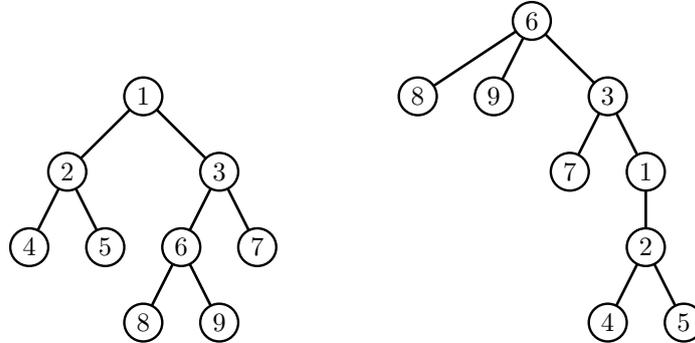
m ответов на запросы.

Пример

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

Задача 15E. Dynamic LCA [1.5 sec, 256 mb]

Наименьший общий предок в дереве для вершин u и v – вершина $lca(u, v)$, которая является предком и u , и v и при этом имеем максимальную глубину. Например, $lca(8, 7) = 3$ в дереве на картинке слева. В дереве можно поменять корень. $chroot(u)$ делает корнем вершину u . Например, после операции $chroot(6)$, как видно на картинке справа, $lca(8, 7) = 6$.



Вам дано дерево с корнем в вершине 1. Напишите программу, отвечающую на запросы $lca(u, v)$ и $chroot(u)$.

Формат входных данных

Во входных данных дано несколько тестов.

Каждый тест начинается с количества вершин в дереве n ($1 \leq n \leq 100\,000$). Следующие $n - 1$ строк содержат пары целых чисел от 1 до n – рёбра дерева. Далее число запросов m ($1 \leq m \leq 200\,000$). Каждая из следующих m строк имеет вид “? u v ” для запроса $lca(u, v)$ или “! u ” для запроса $chroot(u)$.

Последний тест имеет $n = 0$, его обрабатывать не нужно.

Сумма n по всем тестам не более 100 000. Сумма m по всем тестам не более 200 000.

Формат выходных данных

Для каждой операции вида “? u v ” выведите $lca(u, v)$.

Примеры

stdin	stdout
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 15F. LCA Problem Revisited [1.6 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	2
0 1	
2 1	
1 1 0	

Задача 15G. Перестановки [1.2 сек, 256 mb]

Вася выписал на доске в каком-то порядке все числа от 1 по N , каждое число ровно по одному разу. Количество чисел оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая отвечает на вопрос — сколько среди чисел, стоящих на позициях с x по y , по величине лежат в интервале от k до l . Сделайте то же самое.

Формат входных данных

В первой строке лежит два натуральных числа — $1 \leq N \leq 100\,000$ — количество чисел, которые выписал Вася и $1 \leq M \leq 100\,000$ — количество вопросов, которые Вася хочет задать программе. Во второй строке дано N чисел — последовательность чисел, выписанных Васей. Далее в M строках находятся описания вопросов. Каждая строка содержит четыре целых числа $1 \leq x \leq y \leq N$ и $1 \leq k \leq l \leq N$.

Формат выходных данных

Выведите M строк, каждая должна содержать единственное число — ответ на Васин вопрос.

Пример

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

Дополнительные задачи

Задача 15Н. Черепахи и повороты [1 сек, 256 mb]

Для тренировки боевых черепах военные построили прямоугольный полигон размером $W \times H$ клеток. Некоторые клетки проходимы для черепах, а некоторые — нет. Черепахи могут перемещаться только параллельно сторонам полигона. Полигон сконструирован таким образом, что существует единственный способ добраться от любой проходимой клетки до любой другой проходимой клетки, не проходя при этом по одной и той же клетке дважды. Известно, что черепахи очень быстро бегают по прямой, но испытывают трудности при повороте на 90 градусов. Поэтому сложность маршрута определяется как количество поворотов, которое придётся совершить черепахе при переходе от начальной до конечной клетки маршрута. Вы должны написать программу, вычисляющую сложность маршрута по его начальной и конечной клетке.

Формат входных данных

В первой строке через пробел записаны два целых числа H и W — размеры полигона ($1 \leq W \cdot H \leq 100\,000$). Далее задаётся карта полигона — H строк по W символов в каждой. Символ '#' обозначает проходимую клетку, а '.' — непроходимую. В $H+2$ -й строке записано целое число Q — количество маршрутов, для которых нужно посчитать сложность ($1 \leq Q \leq 50\,000$). В каждой из следующих Q строк через пробел записаны четыре целых числа: номер строки и номер столбца начальной клетки маршрута, номер строки и номер столбца конечной клетки маршрута. Гарантируется, что начальная и конечная клетки маршрута являются проходимыми. Строки занумерованы числами от 1 до H сверху вниз, а столбцы — числами от 1 до W слева направо.

Формат выходных данных

Для каждого маршрута выведите в отдельной строке одно число — его сложность.

Примеры

stdin	stdout
5 4	1
.#..	0
###.	2
..##	3
##.	
....	
4	
1 2 2 1	
2 3 4 3	
4 2 3 4	
1 2 4 2	

Задача 15I. LCA-3 [1 sec, 256 mb]

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, $1 \leq n \leq 50\,000$. 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, $1 \leq k \leq 100\,000$. 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

stdin	stdout
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 15J. Дерево [2 сек, 256 mb]

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 . Последующие 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

stdin	stdout
5	1 2 3 1 1
2 1	
3 2	
0 3	
3 3	
2 1	

Задача 15К. Фыр-Фыр-Фенвик [2 сек, 256 mb]

Даны n точек с весами на плоскости. Каждая задаётся тремя числами x_i, y_i, w_i (координаты и вес). Вам нужно обработать m запросов двух типов:

- `get rx ry` – посчитать сумму весов точек, у которых $x_i \leq rx$ и $y_i \leq ry$.
- `change i z` – задать i -й точке новый вес равный z .

Формат входных данных

На первой строке число n ($1 \leq n \leq 100\,000$). На следующих n строках тройки целых чисел x_i, y_i, w_i ($0 \leq x_i, y_i, w_i < 10^9$). Следующая строка содержит количество запросов m ($1 \leq m \leq 300\,000$). На следующих m строках описания запросов в формате `get rx, ry` и `change i z`. Здесь $1 \leq i \leq n$, а остальные числа целые от 0 до 10^9-1 .

Формат выходных данных

Для каждого запроса типа “`get`” выведите одно целое число на отдельной строке — ответ на запрос.

Пример

stdin	stdout
3	110
1 1 10	1210
2 3 100	1010
3 2 1000	
4	
get 2 3	
change 2 200	
get 3 3	
get 3 2	