

Задача А. Размеры поддеревьев

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам дано дерево в корне с вершиной 1.

Поддеревом вершины v называется такое наибольшее множество вершин, в котором путь от каждой вершины до корня содержит вершину v .

Для каждой вершины выведите размер ее поддерева.

Формат входных данных

В первой строке вам дано число n - размер дерева ($2 \leq n \leq 10^6$).

Затем в следующей строке следует описание массива p размера $n - 1$. В нем p_i (где $1 \leq i \leq n - 1$) означает, что в дереве есть ребро между вершинами $i + 1$ и p_i ($1 \leq p_i \leq i$).

Формат выходных данных

Для каждой вершины выведите размер ее поддерева.

Примеры

стандартный ввод	стандартный вывод
6 1 1 2 2 3	6 3 2 1 1 1

Задача В. Предок

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

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

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 100\,000$) — количество вершин в дереве. Во второй строке находятся n чисел, i -е из которых определяет номер непосредственного родителя вершины с номером i . Если это число равно 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.

Примеры

ancestor.in	ancestor.out
6	0
0 1 1 2 3 3	1
5	1
4 1	0
1 4	0
3 6	
2 6	
6 5	

Задача С. Зачёт по географии

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Олимпиадное движение — это очень хорошо, однако не стоит забывать и про школу. Но Гриша именно так и поступил.

Весна. Закончились олимпиады и сборы, пришла пора возвращаться в школу. Гриша учится в Лицее «Бу-Бу-Бу» и, придя в школу, обнаруживает, что ему нужно срочно получать отметки и зачёты по различным предметам, чтобы его не выгнали. Из всех учителей Гриша боится больше всего учителя географии — Дмитрия Константиновича, поэтому желает получить зачёт по его предмету первым. На одной из перемен Гриша подошёл к Дмитрию Константиновичу для обсуждения условий получения зачёта. Гриша предупреждает его, что совершенно не учил географию, поэтому Дмитрий Константинович придумывает другой план.

Дмитрий Константинович решает проверить Гришины знания о географии Лицея! Задание следующее: найти два самых удалённых кабинета в Лицее и посчитать расстояние между ними. Гриша понял, что может не справиться с заданием, поэтому Дмитрий Константинович напомнил, что Лицей представляет собой дерево с N вершинами, где вершинами являются кабинеты, а рёбра — коридоры между ними.

Помогите Грише получить зачёт по географии.

Формат входных данных

В первой строке задаётся одно число N ($2 \leq N \leq 10^5$) — количество кабинетов в Лицее «Бу-Бу-Бу». В следующих $N - 1$ строках задаются по два числа a и b ($1 \leq a, b \leq N$) — коридоры Лицея «Бу-Бу-Бу».

Формат выходных данных

В единственной строке должно содержаться расстояние между двумя самыми удалёнными кабинетами Лицея «Бу-Бу-Бу».

Примеры

стандартный ввод	стандартный вывод
2 1 2	1
3 1 2 1 3	2
4 1 4 1 2 2 3	3

Замечание

P.S. История основана на реальных событиях (почти)

Задача D. Мосты

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

Дан неориентированный граф, не обязательно связный, но не содержащий петель и кратных рёбер. Требуется найти все мосты в нём.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количества вершин и рёбер графа соответственно ($1 \leq n \leq 20\,000$, $1 \leq m \leq 200\,000$).

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leq b_i, e_i \leq n$).

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число b — количество мостов в заданном графе. На следующей строке выведите b целых чисел — номера рёбер, которые являются мостами, **в возрастающем порядке**. Рёбра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Примеры

bridges.in	bridges.out
6 7	1
1 2	3
2 3	
3 4	
1 3	
4 5	
4 6	
5 6	

Задача Е. Точки сочленения

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

Дан неориентированный граф. Требуется найти все точки сочленения в нём.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количества вершин и рёбер графа соответственно ($1 \leq n \leq 20\,000$, $1 \leq m \leq 200\,000$).

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leq b_i, e_i \leq n$).

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число b — количество точек сочленения в заданном графе. На следующей строке выведите b целых чисел — номера вершин, которые являются точками сочленения, в возрастающем порядке.

Примеры

<code>points.in</code>	<code>points.out</code>
6 7	2
1 2	2
2 3	3
2 4	
2 5	
4 5	
1 3	
3 6	

Задача F. Магнитные подушки

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

Город будущего застроен небоскребами, для передвижения между которыми и парковки транспорта многие тройки небоскребов соединены треугольной подушкой из однополярных магнитов. Каждая подушка соединяет ровно 3 небоскреба и вид сверху на нее представляет собой треугольник, с вершинами в небоскребах. Это позволяет беспрепятственно передвигаться между соответствующими небоскребами. Подушки можно делать на разных уровнях, поэтому один небоскреб может быть соединен различными подушками с парами других, причем два небоскреба могут соединять несколько подушек (как с разными третьими небоскребами, так и с одинаковым). Например, возможны две подушки на разных уровнях между небоскребами 1, 2 и 3, и, кроме того, магнитная подушка между 1, 2, 5.

Система магнитных подушек организована так, что с их помощью можно добираться от одного небоскреба, до любого другого в этом городе (с одной подушки на другую можно перемещаться внутри небоскреба), но поддержание каждой из них требует больших затрат энергии.

Требуется написать программу, которая определит, какие из магнитных подушек нельзя удалять из подушечной системы города, так как удаление даже только этой подушки может привести к тому, что найдутся небоскребы из которых теперь нельзя добраться до некоторых других небоскребов, и жителям станет очень грустно.

Формат входных данных

В первой строке входного файла находятся числа N и M — количество небоскребов в городе и количество работающих магнитных подушек соответственно ($3 \leq N \leq 100000$, $1 \leq M \leq 100000$). В каждой из следующих M строк через пробел записаны три числа — номера небоскребов, соединенных подушкой. Небоскребы пронумерованы от 1 до N . Гарантируется, что имеющиеся магнитные подушки позволяют перемещаться от одного небоскреба до любого другого.

Формат выходных данных

Выведите в выходной файл сначала количество тех магнитных подушек, отключение которых невозможно без нарушения сообщения в городе, а потом их номера. Нумерация должна соответствовать тому порядку, в котором подушки перечислены во входном файле. Нумерация начинается с единицы.

Примеры

city.in	city.out
3 1 1 2 3	1 1
3 2 1 2 3 3 2 1	0
5 4 1 2 3 2 4 3 1 2 4 3 5 1	1 4

Задача G. ЧСВ

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Граф Безциклов считает себя очень важной и значимой личностью. Чтобы это доказать, он решил вычислить свою *важность* и *значимость*. *Важность* Графа — количество простых путей в нём, его *значимость* — сумма длин этих путей.

Формат входных данных

Все числа в файле целые.

$0 \leq N \leq 10^5$, $0 \leq M \leq 10^5$ — количество вершин и рёбер.

Затем M пар чисел $1 \leq v_i, u_i \leq N$ — i -е ребро соединяет вершины v_i и u_i .

Граф неориентирован. Гарантируется, что Граф ацикличен.

Формат выходных данных

Вывести важность и значимость Графа.

Пример

стандартный ввод	стандартный вывод
2 1	1
1 2	1

Задача Н. Размещение данных

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

Телекоммуникационная сеть крупной IT-компании содержит n серверов, пронумерованных от 1 до n . Некоторые пары серверов соединены двусторонними каналами связи, всего в сети m каналов. Гарантируется, что сеть серверов устроена таким образом, что по каналам связи можно передавать данные с любого сервера на любой другой сервер, возможно с использованием одного или нескольких промежуточных серверов. Множество серверов A называется отказоустойчивым, если при недоступности любого канала связи выполнено следующее условие. Для любого не входящего в это множество сервера X существует способ передать данные по остальным каналам на сервер X хотя бы от одного сервера из множества A . В условиях показан пример сети и отказоустойчивого множества из серверов с номерами 1 и 4. Данные на сервер 2 можно передать следующим образом. При недоступности канала между серверами 1 и 2 — с сервера 4, при недоступности канала между серверами 2 и 3 — с сервера 1. На серверы 3 и 5 при недоступности любого канала связи можно по другим каналам передать данные с сервера 4.

В ходе проекта группе разработчиков компании необходимо разместить свои данные в сети. Для повышения доступности данных и устойчивости к авариям разработчики хотят продублировать свои данные, разместив их одновременно на нескольких серверах, образующих отказоустойчивое множество. Чтобы минимизировать издержки, необходимо выбрать минимальное по количеству серверов отказоустойчивое множество. Кроме того, чтобы узнать, насколько гибко устроена сеть, необходимо подсчитать количество способов выбора такого множества, и поскольку это количество способов может быть большим, необходимо найти остаток от деления этого количества способов на число $10^9 + 7$. Требуется написать программу, которая по заданному описанию сети определяет следующие числа: k — минимальное количество серверов в отказоустойчивом множестве серверов, c — остаток от деления количества способов выбора отказоустойчивого множества из k серверов на число $10^9 + 7$.

Формат входных данных

Первая строка входного файла содержит целые числа n и m — количество серверов и количество каналов связи соответственно ($2 \leq n \leq 200000, 1 \leq m \leq 200000$). Следующие m строк содержат по два целых числа и описывают каналы связи между серверами. Каждый канал связи задается двумя целыми числами: номерами серверов, которые он соединяет. Гарантируется, что любые два сервера соединены напрямую не более чем одним каналом связи, никакой канал не соединяет сервер сам с собой, и для любой пары серверов существует способ передачи данных с одного из них на другой, возможно с использованием одного или нескольких промежуточных серверов.

Формат выходных данных

Выведите два целых числа, разделенных пробелом: k — минимальное число серверов в отказоустойчивом множестве серверов, c — количество способов выбора отказоустойчивого множества из k серверов, взятое по модулю $10^9 + 7$.

Примеры

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

Замечание

В приведенном примере отказоустойчивыми являются следующие множества $\{1, 3\}, \{1, 4\}, \{1, 5\}$.

Задача I. Спасение цивилизации

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

Астрономы засекли опасный метеорит, приближающийся на огромной скорости к планете. По данным ученых метеорит населен опасными монстрами. Известно, что города планеты можно представить в виде дерева (*связанного неориентированного графа без циклов*), где вершины — города планеты, а ребра — двусторонние дороги между ними.

К счастью, астрономы смогли точно определить, что метеорит упадет в город R , поэтому теперь появился шанс уничтожить некоторые дороги, чтобы помешать монстрам заполнить хотя бы часть планеты.

Ученые выяснили, что для сохранения цивилизации нужно спасти хотя бы K городов.

Про каждую дорогу известно, возможно ли ее уничтожить. К сожалению, до вторжения остается совсем немного времени, а перекрытие любой дороги требует времени. Помогите ученым понять, смогут ли они спасти цивилизацию, и какое минимальное количество дорог нужно уничтожить, чтобы по оставшимся дорогам монстры не смогли добраться из города R хотя бы до K городов планеты.

Формат входных данных

В первой строке через пробел вводятся два целых числа N , R и K ($2 \leq N \leq 100\,000, 1 \leq R, K \leq N$) — количество городов на планете, город, в который попадет метеорит, и количество городов, которые необходимо спасти, соответственно.

В каждой из следующих $N - 1$ -й строк вводится по три целых числа v, u, w , означающих что между v и u существует двусторонняя дорога, которую можно уничтожить, если $w = 1$ и нельзя, если $w = 0$.

Гарантируется, что введенные ребра образуют дерево.

Формат выходных данных

Если спасти K городов невозможно, то выведите -1 , иначе выведите минимальное число дорог, которые необходимо удалить.

Пример

стандартный ввод	стандартный вывод
7 2 3 1 3 0 1 7 0 2 1 1 2 5 0 5 6 1 2 4 1	1