Задача A. Мосты

Имя входного файла: `стандартный ввод`
Имя выходного файла: `стандартный вывод`
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 ($1 \leqslant n \leqslant 20\,000$, $1 \leqslant m \leqslant 200\,000$).

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

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.

Пример

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

Задача B. Точки сочленения

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

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 ($1 \leqslant n \leqslant 20\,000$, $1 \leqslant m \leqslant 200\,000$).

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

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.

Пример

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

Задача C. Компоненты реберной двусвязности

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Компонентой реберной двусвязности графа $\langle V, E \rangle$ называется подмножество вершин $S \subset V$, такое что для любых различных u и v из этого множества существует не менее двух реберно не пересекающихся путей из u в v.

Дан неориентированный граф. Требуется выделить компоненты реберной двусвязности в нем.

Формат входных данных

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

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

Формат выходных данных

В первой строке выходного файла выведите целое число k — количество компонент реберной двусвязности графа.

Во второй строке выведите n натуральных чисел $a_1, a_2, \ldots, a_n$, не превосходящих k, где a_i — номер компоненты реберной двусвязности, которой принадлежит i-я вершина.

Компоненты требуется нумеровать в порядке возрастания минимального номера вершины, входящей в компоненту.

Пример

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

Задача D. Компоненты вершинной двусвязности

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Компонентой вершинной двусвязности графа $\langle V, E\rangle$ называется максимальный по включению подграф (состоящий из вершин и ребер), такой что любые два ребра из него лежат на вершинно простом цикле.

Дан неориентированный граф без петель. Требуется выделить компоненты вершинной двусвязности в нем.

Формат входных данных

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

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

Формат выходных данных

В первой строке выходного файла выведите целое число k — количество компонент вершинной двусвязности графа.

Во второй строке выведите m натуральных чисел $a_1, a_2, \ldots, a_m$, не превосходящих k, где a_i — номер компоненты вершинной двусвязности, которой принадлежит i-е ребро. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Компоненты требуется нумеровать в порядке возрастания минимального номера ребра, входящего в компоненту.

Пример

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

Задача E. Магнитные подушки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
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 ($3 \leqslant N \leqslant 100000$, $1 \leqslant M \leqslant 100000$). 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

стандартный ввод	стандартный вывод
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 F. Здоровье Графа

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

Этично ли удалять рёбра у связанного графа?

Граф Безциклов решил проверить своё здоровье. Он хочет проверить, что все его рёбра достаточно крепко держатся в нём. Для этого он хочет посчитать *устойчивость* некоторых из них. *Устойчивостью* ребра называется количество простых путей, проходящих через это ребро.

Формат входных данных

Все числа в файле целые.
$0 \leqslant N \leqslant 10^5$, $0 \leqslant M \leqslant 10^5$ — количество вершин и рёбер.
Затем M пар чисел $1 \leqslant v_i, u_i \leqslant N$ — i-ое ребро соединяет вершины v_i и u_i.
$0 \leqslant Q \leqslant 10^5$ — количество запросов.
Затем Q чисел $1 \leqslant e_i \leqslant M$.
Граф неориентирован. Гарантируется, что Граф ацикличен.

Формат выходных данных

Для i-ого запроса вывести устойчивость e_i-ого ребра.

Примеры

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

Задача G. Выбор вершин взвешенного дерева

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

Дан граф, являющийся деревом. В вершинах графа написаны целые числа. Множество вершин графа называется *допустимым*, если никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа. Для каждого такого множества вычислим сумму чисел, написанных в его вершинах. Какова максимальная из этих сумм?

Формат входных данных

Граф в этой задаче задан в виде *корневого дерева*. В графе выделена вершина — *корень дерева*. Для каждой вершины i, не являющейся корнем, задан номер вершины-предка p_i в корневом дереве. Дерево, заданное таким образом, состоит из рёбер $i - p_i$ для всех вершин i, кроме корня.

В первой строке входного файла записано целое число n — количество вершин в графе ($1 \leqslant n \leqslant 100$). В следующих n строках задан граф. В i-й из этих строк записаны через пробел два целых числа p_i и q_i; здесь p_i — номер вершины-предка i-ой вершины, а q_i — число, записанное в этой вершине. Для корня дерева $p_i = 0$; для всех остальных вершин $1 \leqslant p_i \leqslant n$. Числа q_i не превосходят по модулю 10 000.

Гарантируется, что заданный во входном файле граф является деревом.

Формат выходных данных

В первой строке выходного файла выведите одно число — максимальную сумму чисел в допустимом множестве.

Примеры

стандартный ввод	стандартный вывод
5 0 1 1 2 1 3 2 4 3 5	10
6 5 8 6 0 5 -1 1 1 0 3 1 2	8

Задача H. Максимальная тройка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Дано дерево на n вершинах. Требуется выбрать из них три так, чтобы сумма расстояний между ними была максимальна.

Формат входных данных

Первая строка каждого теста содержит натуральное число n — количество вершин в дереве ($3 \leqslant n \leqslant 1\,000\,000$). Следующие $n-1$ строк содержат по 2 натуральных числа v, u и описывают ребро дерева, соединяющее две вершины v и u ($1 \leqslant v, u \leqslant n$).

Формат выходных данных

Выведите единственное число – максимальную сумму расстояний.

Примеры

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

Задача I. Размещение данных

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
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.

На рис. 1 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.

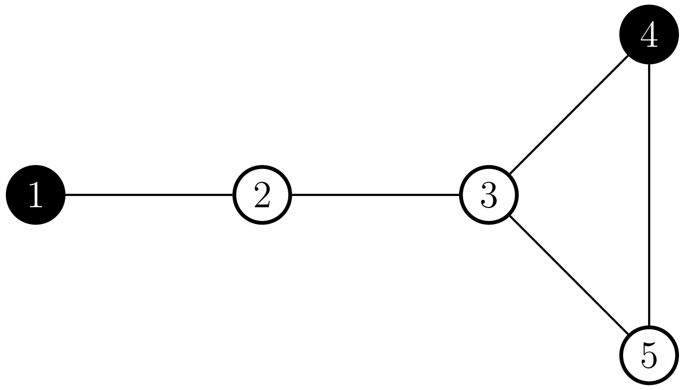

Рис. 1: Пример сети и отказоустойчивого множества серверов.

В рамках 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 ($2 \leqslant n \leqslant 200\,000$, $1 \leqslant m \leqslant 200\,000$). 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 $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
5 5 1 2 2 3 3 4 3 5 4 5	2 3

Задача J. Минимизация мостов

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

Добавить в граф $G = \langle V, E \rangle$ (возможно несвязный, с петлями и кратными рёбрами) ровно одно ребро, так чтобы количество мостов в данном графе стало минимально возможным.

Напомним, что мостом в графе называется такое ребро, удаление которого увеличивает число компонент связности графа.

Формат входных данных

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

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается двумя натуральными числами v_i , u_i – номерами концов ребра ($1 \leqslant v_i, u_i \leqslant n$).

Формат выходных данных

Выведите наименьшее число мостов, которое можно получить добавлением ровно одного ребра.

Пример

<table>
<tr><th>стандартный ввод</th><th>стандартный вывод</th></tr>
<tr><td><pre>6 7
1 2
2 3
3 4
1 3
4 5
4 6
5 6</pre></td><td><pre>0</pre></td></tr>
</table>

Задача K. Миссия выполнима!

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мегабайта

Это упрощенная версия задачи с дополнительными ограничениями

Агент Итан получил задания по сбору секретных документах в лаборатории. Лаборатория состоит из n комнат, соединенных $n-1$ коридорами. Каждый коридор соединяет две различные комнаты, из любой комнаты есть путь до любой другой, переход по коридору i, соединяющему комнаты a_i, b_i занимает $c_i \geqslant 0$ времени.

Когда Итан проник в комнату с номером x, сработала сигнализация. По заданию ему нужно посетить еще k любых различных комнат в лаборатории (включая x, она уже считается посещенной). Для оценки возможности завершения миссии, необходимо вычислить минимальное время, необходимое для посещения k любых различных комнат в лаборатории (включая x, она уже считается посещенной). Порядок посещения комнат и комната, в которой посещение закончится не имеют значения. Возвращаться в комнату x не обязательно.

Ваша миссия, если Вы возьметесь за ее выполнение, – вычислить искомое минимальное время.

Формат входных данных

Первая строка содержит целые числа n, k и x ($1 \leqslant k, x \leqslant n \leqslant 10^4$).

Следующие $n-1$ строк описывают коридоры тремя числами a_i, b_i, c_i ($1 \leqslant a_i, b_i \leqslant n; 0 \leqslant c_i \leqslant 10^4$). Это означает, что переход между комнатами a_i, b_i занимает c_i времени.

Дополнительные ограничения: $k \leqslant \min(n, 100)$, каждая комната соединена коридором напрямую не более чем с 3 комнатами.

Формат выходных данных

Выведите единственное целое число – минимальное время, необходимое для посещения k различных комнат.

Примеры

стандартный ввод	стандартный вывод
10 8 3 1 3 3 3 2 5 6 4 5 1 8 3 9 1 2 9 10 2 3 7 10 6 7 1 7 5 1	32
3 1 1 1 2 4 2 3 0	0

Задача L. Странные деревья

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

У вас есть связный неориентированный граф с n вершинами и m рёбрами. Каждое ребро имеет ассоциированный счётчик, изначально равный 0. За одну операцию вы можете выбрать произвольное остовное дерево и добавить любое значение v ко всем рёбрам этого остовного дерева.

Определите, возможно ли сделать каждый счётчик равным целевому значению x_i по модулю простого числа p, и предоставьте последовательность операций, достигающую этого.

Формат входных данных

Первая строка содержит три целых числа n, m, и p — количество вершин, количество рёбер и простое модуль ($1 \leqslant n \leqslant 500$; $1 \leqslant m \leqslant 1000$; $2 \leqslant p \leqslant 10^9$, p простое число).

Далее следуют m строк, содержащих по три целых числа u_i, v_i, x_i каждое — две конечные точки i-го ребра и целевое значение счётчика этого ребра ($1 \leqslant u_i, v_i \leqslant n$; $0 \leqslant x_i < p$; $u_i \neq v_i$).

Граф связный. Петель нет, но могут быть несколько рёбер между одной и той же парой вершин.

Формат выходных данных

Если целевые значения счётчиков недостижимы, выведите `-1`.

В противном случае, выведите t — количество операций, за которыми следуют t строк, описывающих последовательность операций. Каждая строка начинается с целого числа v ($0 \leqslant v < p$) — увеличение счётчика для этой операции. Затем, на той же строке, следуют $n-1$ целых чисел e_1, e_2, $\ldots e_{n-1}$ ($1 \leqslant e_i \leqslant m$) — рёбра остовного дерева.

Количество операций t не должно превышать $2m$. Вам не нужно минимизировать t. Любой правильный ответ в пределах ограничения $2m$ принимается. Вам разрешено повторять остовные деревья.

Примеры

стандартный ввод	стандартный вывод
3 3 101 1 2 30 2 3 40 3 1 50	5 60 2 1 51 2 1 50 3 2 20 3 1 81 3 2
2 2 37 1 2 8 1 2 15	3 23 1 22 1 15 2
5 4 5 1 3 1 2 3 2 2 5 3 4 1 4	-1