Задача A. Дерево доминаторов

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	512 мегабайт

Дан ориентированный граф с n вершинами и m ребрами. Пусть S - это вершина 1. Вершина A *доминирует* над вершиной B, если любой путь из S в B проходит через A. В частности, вершина S доминирует над всеми вершинами. *Непосредственный доминатор* вершины B (далее $idom(v)$) - это самый близкий доминатор к вершине B на произвольно выбранном пути из S в B. Далее непосредственный доминатор обозначается за idom(B).

Формат входных данных

Вход содержит несколько тестовых примеров. Каждый тест начинается со строки, содержащей два целых числа n и m ($2 \leqslant n \leqslant 2 \cdot 10^5; 1 \leqslant m \leqslant 2 \cdot 10^5$) — количество вершин и ребер в графе. Следующие m строк описывают ребра. Описание состоит из пары чисел u_i и v_i ($1 \leqslant u_i, v_i \leqslant n; u_i \neq v_i$), означающих, что i-е ребро начинается в вершину u_i и заканчивается в вершине v_i. Вершины пронумерованы от 1 до n.

Гарантированно, что между каждой парой вершин не более одного ребра в каждом направлении.

Каждый тестовый пример завершается пустой строков. Сумма n по всем тестовым примерам не превосходит $2 \cdot 10^5$. Аналогично для m: сумма m по всем тестовым примерам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого тестового примера выведите n чисел в одну строку через пробел. i-е число в строке должно равняться $idom(i)$. Если вершина i не достижима из вершины 1, считайте $idom(i)$ равным -1. Также $idom(1)$ считайте равным 1.

Примеры

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

Задача B. Два пути

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

В Байтландии города соединены односторонними дорогами. Дорожная система Байтландии обладает одним интересным свойством: если вы выехали из города по какой-то дороге, вы не сможете вернуться назад. Иными словами, структура дорог может быть описана ориентированным ациклич-ным графом.

Такая особенность создает некоторые проблемы. Например, до некоторых городов нельзя добраьтся даже из столицы Байтландии. Бывает и хуже: дороги часто закрываются на ремонт. В таком случае до некоторых городов нельзя будет добраться, даже если до них можно было доехать ранее.

Байтазер живет в столице и часто путешествует в другие города Байтландии. Для каждого города C он хотел бы знать, существуют ли два пути из столицы в C, которые не имеют общих дорог. Если это правда (или C является столицей), Байтазаер знает, что путешествие до C всегда возможно, даже если какая-то из дорог закрыта на ремонт. Помогите Байтазару найти все города, до которых можно добраться из столицы, даже если какая-нибудь одна дорога закроется на ремонт.

Формат входных данных

Первая строка содержит два числа n, m ($1 \leqslant n \leqslant 200\,000$, $1 \leqslant m \leqslant 500\,000$) – количество городов и дорог в Байтландии. Города пронумерованы числами $1, 2, \ldots, n$, столица имеет номер 1. Следующие m строк описывают дорог: i-я из них содержит два числа u_i, v_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$) означающие, что i-я односторонняя дорога начинается в городе u_i и заканчивается в городе v_i.

Формат выходных данных

В первой строке выведите количество городов, до которых можно добраться до столицы, даже если одна из дорог закроется на ремонт. В следующей строка выведите номера этих городов по возрастанию, разделенных пробелами.

Примеры

стандартный ввод	стандартный вывод
7 9 1 2 1 3 3 4 4 5 2 4 2 5 5 6 5 7 5 7	4 1 4 5 7

Задача C. Полезные дороги

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	6 секунд
Ограничение по памяти:	512 мегабайт

В столице Берляндии n перекрестков и m дорог, соединяющих их. По дорогам можно передвигаться лишь в одном направлении. Как вы догадываетесь, в Берляндии две беды: дураки и дороги.

До выборов мэра остался один месяц, поэтому сейчас самое время для текущего правительства делать город лучше. Чтобы показать, что они заботятся об инфраструктуре и бюджете, правительство решило починить только *полезные* дороги.

Текущий мэр считает дорогу от перекрестка u к перекрестку v полезной, если существует простой путь, содержащий дорогу (u, v), начинающийся в мэрии и заканчивающийся в каком-либо перекрестке. Путь называется простым, если никакой перекресток на этом пути не повторяется. Мэрия находится на перекрестке номер 1.

Помогите министерству транспорта найти все полезные дороги в городе.

Формат входных данных

Вход содержит несколько тестовых примеров. Каждый тест начинается со строки, содержащей два целых числа n и m ($2 \leqslant n \leqslant 2 \cdot 10^5; 1 \leqslant m \leqslant 2 \cdot 10^5$) — количество перекрестков и дорог в городе. Следующие m строк описывают дороги. Описание состоит из пары чисел u_i и v_i ($1 \leqslant u_i, v_i \leqslant n; u_i \neq v_i$), означающих, что i-я дорога начинается в перекрестке u_i и заканчивается в перекрестке v_i. Перекрестки пронумерованы от 1 до n. Мэрия находится на перекрестке номер 1.

Гарантированно, что между каждой парой перекрестков не более одной дороги в каждом направлении.

Каждый тестовый пример завершается пустой строков. Сумма n по всем тестовым примерам не превосходит $2 \cdot 10^5$. Аналогично для m: сумма m по всем тестовым примерам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждоого тестового примера выведите две строки. В первой строке должно находиться количество полезных дорог. Во второй строке находятся индексы полезных дорог в возрастающем порядке. Дороги пронумерованы от 1 до m в том же порядке, как они идут во входном файле. Если полезных дорог нет, оставьте вторую строку пустой.

Примеры

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

Задача D. Увеличение стоимости

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Берляндия состоит из n городов, пронумерованных от 1 до n. Город номер 1 является столицей Берляндии. Также в Берляндии m двунаправленных дорог между некоторыми городами. Разные дороги пересекаются только в городах. Между каждой парой городов не больше одной дороги, и никакая дорога не соединяет город с собой. Если вы передвигаетесь по j-й дороге в любом направлении, вы платите пошлину c_j. **По заданным m дорогам можно добраться до каждого города от столицы.**

Вы руководите компанией, занимающейся доставкой, ваш главный офис находится в столице. Ваша компания доставляет различные посылки в каждый город Берляндии, поэтому для каждого города вы выбрали путь, минимизирующий суммарную пошлину всех дорог в этом пути. Обозначим суммарную пошлину на пути до вершины k как d_k.

Государство решило выбрать **ровно одну** дорогу (вы не знаете, какую именно) и увеличить пошлину за ее использование. Для каждой дороги вы хотели бы знать, сколько городов будут затронуты этим увеличением. Город k считается затронутым изменением, если после увеличения пошлины за использование дороги, вы не сможете выбрать путь суммарной стоимости d_k.

Формат входных данных

Первая строка содержит два числа n и m: количество городов и дорог в Берляндии ($2 \leqslant n \leqslant 2 \cdot 10^5$, $n - 1 \leqslant m \leqslant 2 \cdot 10^5$).

Каждая из следующих m дорог содержит по три целых числа: u_j, v_j, и c_j ($1 \leqslant u_j, v_j \leqslant n$, $1 \leqslant c_j \leqslant 10^9$), означающих, что j-я дорога соединяет города u_j и v_j, а пошлина за ее использование равна c_j.

Между каждой парой городов не более одной дороги, никакая дорога не соединяет город с собой. Гарантируется, что из столицы можно добраться до любого города по заданным дорогам.

Формат выходных данных

Выведите m чисел, каждое в отдельной строке. Число в строке j должно быть равно количеству городов, затронутых увеличением пошлины j-й дороги.

Примеры

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

Задача E. Ski Resort

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	6 секунд
Ограничение по памяти:	1024 мегабайта

As an enterprising owner of a world-renowned ski resort in the US, you would like to increase your sales by stocking snack stands at key locations throughout your estate.

The ski resort is built on a mountain. A ski lift can take a guest to the top of the mountain. From there they can ski to a number of locations throughout the mountain.

There are n areas on the mountain. The areas are labeled $1 \ldots n$. The top of the mountain is area 1. Areas are connected with ski runs that are strictly downhill. In other words, it is not possible to return to an area after leaving it without taking the ski lift. Every area (including area 1) has exactly one snack stand.

As the owner of the resort, you want to know how effectively you can distribute snacks among snack stands to better serve your guests (and make more money). To this end you would like to run a survey, and analyze the result with a number of independent queries. Each guest in the survey has a favorite snack, and a list of favorite areas that they like to visit. You would like to know how to best stock your snack stands with their favorite snack.

Each query is a set of a guest's favorite areas, and a number k. You would like to know how many ways you can distribute this guest's favorite snack to exactly k snack stands on the mountain such that the snack stands meet a few conditions:

1. For each of this guest's favorite areas, over all sequences of ski runs to reach that area from the top of the mountain, there must be exactly one snack stand with the guest's favorite snack (In other words, they must not have a choice of more than one snack stand where their snack is available.)

2. Each of the k snack stands stocked with this guest's favorite snack must be on some sequence of ski runs from the top of the mountain to some area in the query set.

Формат входных данных

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. The first line of input will contain three space-separated integers n, m, and q, where n ($1 \leqslant n \leqslant 10^5$) is the number of areas on the mountain, m ($1 \leqslant m \leqslant n + 50$) is the number of runs, and q ($1 \leqslant q \leqslant 10^5$) is the number of queries.

The next m lines each contain two integers x and y ($1 \leqslant x, y \leqslant n, x \neq y$). This represents a ski run from area x to area y. There will be at most one run between any two areas. It will be possible to reach each area from area 1 by some chain of ski runs.

The next q lines are each a sequence of space-separated integers, starting with k and a, which are followed by a integers i. Here, k ($1 \leqslant k \leqslant 4$) represents the number of snack stands to stock with this guest's favorite snack, a ($1 \leqslant a \leqslant n$) represents the number of areas in the query set, and the a integers i ($1 \leqslant i \leqslant n$) are the labels of the areas in the query set. In any given query, no integer i will be repeated.

The sum of all a's for all queries will not exceed $100,000$.

Формат выходных данных

Output q integers, each on its own line with no blank lines in between. These represent the number of ways to select snack stands to stock for each query, in the order that they appear in the input. Two ways are considered different if an area is selected in one configuration but not the other.

Примеры

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

Задача F. Олимпиады и другие активности

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

В 2030 году в Верхневолжской Школе Электроники все студенты самоидентифицируются либо как кошкодевочки, либо как кошкомальчики.

Филип хочет разбить их на пары для соревнования "по программированию". По причинам, никак не связанным ни с какими правительствами, каждая пара должна состоять из кошкомальчика и кошкодевочки. Студенты не делятся на пары сами, а предлагают Филипу возможные пары и доверяют ему итоговое разделение. Открытое признание в том, что вы хотите быть в паре со студентом, который идентифицируется так же, как и вы, в ВШЭ очень серьёзно наказывается, поэтому все предложенные пары содержат ровно одну кошкодевочку. Разумеется, внешне кошкомальчики и кошкодевочки ничем не отличаются и Филип не знает, кто есть кто.

Филип хочет, чтобы пар было как можно больше. Однако, Филипу в любое время нужны 2 свободных студента, чтобы пойти с ними по барам. Эти двое будут находится в состоянии, в котором ни в каких соревнованиях участвовать не получится.

Но Филип не хочет, чтобы кто-то заметил, что команд меньше, чем должно быть, поэтому он хочет узнать количество способов выбрать пару студентов так, чтобы наибольшее количество пар без них совпадало с наибольшим количеством пар с ними.

Формат входных данных

В первой строке находятся два целых числа: количество студентов n ($2 \leqslant n \leqslant 2 \cdot 10^5$) и количество предложенных пар m ($0 \leqslant m \leqslant 2 \cdot 10^5$).

В каждой из следующих m строк находятся по 2 целых числа, описывающие предложенные пары — a и b ($1 \leqslant a, b \leqslant n, a \neq b$).

Гарантируется, что все предложения уникальны, то есть для любых a и b есть не более одного предложения объединить их в пары.

Формат выходных данных

В единственной строке выведите количество возможных пар собутыльников Филипа.

Примеры

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