

Задача А. 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 числа n и m ($2 \leq n \leq 2 \cdot 10^5$; $1 \leq m \leq 2 \cdot 10^5$) — количество вершин и ребер в графе. Следующие m строк описывают ребра. Описание состоит из пары чисел u_i и v_i ($1 \leq u_i, v_i \leq n$; $u_i \neq v_i$), означающих, что i -е ребро идёт из вершины u_i в вершину v_i . Вершины пронумерованы от 1 до n . Между каждой парой вершин не более одного ребра в каждом направлении. Каждый тестовый пример завершается пустой строкой.

Сумма n по всем тестовым примерам не превосходит $2 \cdot 10^5$.

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 В. 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 ациклическим графом.

Такая особенность создает некоторые проблемы. Например, до некоторых городов нельзя добраться 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), Байтазер 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 ($1 \leq n \leq 200\,000$, $1 \leq m \leq 500\,000$) – количество городов и дорог в Байтландии. Города пронумерованы числами $1, 2, \dots, n$, столица имеет номер 1. Следующие m строк описывают дорог: i -я из них содержит два числа u_i, v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) 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 строке 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 С. 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 ($2 \leq n \leq 2 \cdot 10^5; 1 \leq m \leq 2 \cdot 10^5$) — количество перекрестков и дорог в городе. Следующие m строк описывают дороги. Описание состоит из пары чисел u_i и v_i ($1 \leq u_i, v_i \leq n; u_i \neq v_i$), 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.

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

Каждый тестовый пример завершается пустой строкой. 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 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 двусторонних 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 ($2 \leq n \leq 2 \cdot 10^5$, $n - 1 \leq m \leq 2 \cdot 10^5$).

Каждая из следующих m дорог содержит по три целых числа: u_j , v_j , и c_j ($1 \leq u_j, v_j \leq n$, $1 \leq c_j \leq 10^9$), 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	5
1 2 2	1
2 3 1	0
3 4 7	0
4 5 4	1
5 2 4	1
4 6 4	

Задача Е. Горнолыжный курорт

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

Как владелец элитного горнолыжного курорта вы хотите сделать его еще более популярным, улучшив систему кафе на курорте, сделав список блюд в них более подходящим для каждого гостя.

Сейчас на курорте есть n смотровых площадок, рядом с каждой из которых есть ровно одно кафе, несколько горнолыжных трасс, соединяющих их, и один подъемник, который может доставить гостей курорта к самой высокой смотровой площадке под номером 1. При этом все горнолыжные трассы идут строго вниз, т.е. вы не можете вернуться на смотровую площадку, после того, как вы уехали из нее, не используя подъемник.

Для того, чтобы сделать отдых ваших гостей еще лучше, вы провели опрос и для каждого гостя узнали его любимое блюдо, а также список смотровых площадок, на которых ему нравится бывать.

Каждый запрос состоит из списка любимых смотровых площадок гостя и числа k . Вы хотите узнать количество способов добавить его любимое блюдо в меню **ровно** k кафе так, чтобы были выполнены следующие условия:

1. Для каждой из любимых смотровых площадок гостя, для каждого из возможных путей, по которым можно добраться до нее от самой вершины, должно быть ровно одно кафе с его любимым блюдом. (Другими словами у него не должно быть выбора, где именно ему съесть его любимое блюдо после того, как он зафиксировал маршрут от вершины горы до одной из его любимых смотровых площадок)
2. Все k кафе с его любимым блюдом должны располагаться на пути от вершины горы до одной из его любимых смотровых площадок

Формат входных данных

В первой строке ввода содержатся три числа $1 \leq n \leq 10^5$, $1 \leq m \leq n + 50$ и $1 \leq q \leq 10^5$ - количество смотровых площадок, горнолыжных трасс и запросов соответственно.

В каждой из следующих m строк содержится два числа x и y ($1 \leq x, y \leq n, x \neq y$) - обозначающих горнолыжную трассу, проложенную от площадки x до площадки y . Гарантируется, что между двумя площадками проложено не более одной трассы. Также гарантируется, что по цепочке горнолыжных трасс можно добраться от площадки 1 до любой другой площадки.

В следующих q строках идут описания запросов. В каждой строке запроса сначала идут числа k и a - количество кафе, в которые нужно добавить любимое блюдо очередного гостя и количество любимых смотровых площадок этого гостя. Далее в этой же строке идут a чисел с номерами любимых площадок этого гостя. Гарантируется, что сумма a по всем запросам не превосходит 10^5

Формат выходных данных

Выведите q строк, каждая из которых содержит одно число - количество способов выбрать кафе, в которые добавить блюдо очередного гостя так, чтобы все условия были соблюдены.

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 находиться 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 ($2 \leq n \leq 2 \cdot 10^5$) и количество предложенных пар m ($0 \leq m \leq 2 \cdot 10^5$).

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

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