Задача A. Topsort

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

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входных данных

В первой строке входного файла даны два целых числа N и M ($1 \leqslant N \leqslant 100\,000, 0 \leqslant M \leqslant 100\,000$) — количества вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходных данных

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести «`-1`».

Пример

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

Задача B. Топологическая сортировка

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

Задан ориентированный ациклический граф с n вершинами и m ребрами. Также задана перестановка вершин графа. Необходимо проверить, является ли данная перестановка топологической сортировкой.

Формат входных данных

В первой строке даны два числа n и m — количество вершин и ребер в графе соответственно ($1 \leqslant n, m \leqslant 10^5$). В следующих m строках заданы пары чисел u_i, v_i, означающие, что в графе есть ребро из вершины u_i в вершину v_i. В последней строке задана перестановка из n элементов.

Формат выходных данных

Выведите «YES» (без кавычек), если данная перестановка является топологической сортировкой и «NO» в противном случае.

Примеры

стандартный ввод	стандартный вывод
3 3 2 3 1 3 1 2 2 1 3	NO
3 3 3 2 1 2 3 1 3 1 2	YES

Задача C. Конденсация графа

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

Вам задан ориентированный граф с N вершинами и M ребрами ($1 \leqslant N \leqslant 200\,000$, $1 \leqslant M \leqslant 200\,000$). Найдите компоненты сильной связности заданного графа и топологически отсортируйте его конденсацию.

Формат входных данных

Граф задан во входном файле следующим образом: первая строка содержит числа N и M. Каждая из следующих M строк содержит описание ребра – два целых числа из диапазона от 1 до N – номера начала и конца ребра.

Формат выходных данных

На первой строке выведите число K – количество компонент сильной связности в заданном графе. На следующей строке выведите N чисел – для каждой вершины выведите номер компоненты сильной связности, которой принадлежит эта вершина. Компоненты сильной связности должны быть занумерованы таким образом, чтобы для любого ребра номер компоненты сильной связности его начала не превышал номера компоненты сильной связности его конца.

Пример

стандартный ввод	стандартный вывод
10 19 1 4 7 8 5 10 8 9 9 6 2 6 6 2 3 8 9 2 7 2 9 7 4 5 3 6 7 3 6 7 10 8 10 1 2 9 2 7	2 1 2 2 1 1 2 2 2 2 1

Задача D. Кратчайший путь

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

Дан ориентированный ациклический взвешенный граф. Требуется найти в нем кратчайший путь из вершины s в вершину t.

Формат входных данных

Первая строка входного файла содержит четыре целых числа n, m, s и t — количество вершин, дуг графа, начальная и конечная вершина соответственно.

Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается тремя натуральными числами b_i, e_i и w_i — началом, концом и длиной дуги соответственно ($1 \leqslant b_i, e_i \leqslant n$, $|w_i| \leqslant 1\,000$).

Входной граф не содержит циклов и петель.

$1 \leqslant n \leqslant 100\,000$, $0 \leqslant m \leqslant 200\,000$.

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — длину кратчайшего пути из s в t.

Если пути из s в t не существует, выведите «`Unreachable`».

Пример

стандартный ввод	стандартный вывод
`2 1 1 2` `1 2 -10`	`-10`

Задача E. Сумма длин путей

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

Дано дерево на n вершинах. На каждом ребре написан его вес. Требуется посчитать сумму взвешенных длин всех путей в данном дереве. Пути $\langle v, u \rangle$ и $\langle u, v \rangle$ считаются различными.

Формат входных данных

Первая строка каждого теста содержит натуральное число n — количество вершин в дереве ($1 \leqslant n \leqslant 100\,000$). Следующие $n - 1$ строк содержат по 3 натуральных числа v, u, w и описывают ребро дерева, соединяющее две вершины v и u и имеющее вес w ($1 \leqslant v, u \leqslant n$, $0 \leqslant w \leqslant 10^6$).

Формат выходных данных

Выведите единственное число — суммарную длину всех путей в дереве.

Примеры

стандартный ввод	стандартный вывод
3 1 2 1 1 3 3	16
7 1 2 1 3 4 1 7 6 1 7 5 1 7 1 1 1 4 1	92

Задача F. Сумма длин путей 2

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

Дано дерево на n вершинах. На каждом ребре написан его вес. Требуется для каждого v посчитать сумму взвешенных длин всех путей в данном дереве, исходящих из v. Пути $\langle v, u \rangle$ и $\langle u, v \rangle$ считаются различными.

Формат входных данных

Первая строка каждого теста содержит натуральное число n — количество вершин в дереве ($1 \leqslant n \leqslant 100\,000$). Следующие $n - 1$ строк содержат по 3 натуральных числа v, u, w и описывают ребро дерева, соединяющее две вершины v и u и имеющее вес w ($1 \leqslant v, u \leqslant n$, $0 \leqslant w \leqslant 10^6$).

Формат выходных данных

Выведите n чисел — требуемое в условии.

Примеры

стандартный ввод	стандартный вывод
3 1 2 1 1 3 3	4 5 7
7 1 2 1 3 4 1 7 6 1 7 5 1 7 1 1 1 4 1	9 14 17 12 15 15 10

Задача G. Расстановка чисел

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

Вам даны N переменных $a_1, a_2, \ldots, a_N$. Также про некоторые из этих переменных известно, что одна из них делится на другую.

Пусть мы присвоили этим переменным целые значения так, что все утверждения выполняются.

Требуется найти максимально возможное число **различных** значений среди этих переменных.

Формат входных данных

В первой строке входного файла задано два целых числа число N и M — количество переменных и количество отношений делимости ($1 \leqslant N \leqslant 10^4, 1 \leqslant M \leqslant 10^5$).

Далее следуют M пар чисел i и j ($1 \leqslant i, j, \leqslant N$) задающих, что a_i делится на a_j.

При этом возможны как повторяющиеся утверждения, так и тривиальные (в которых $i = j$).

Формат выходных данных

Выведите одно число — наибольшее количество различных целых чисел в наборе a_i, удовлетворяющем всем утверждениям.

Пример

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

Замечание

В примере к задаче можно, например, взять $a_1 = 0$, $a_2 = -4$, $a_3 = 2$ и $a_4 = 3$.

Задача H. Производство деталей

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

Предприятие «Авто-2010» выпускает двигатели для известных во всём мире автомобилей. Двигатель состоит ровно из n деталей, пронумерованных от 1 до n, при этом деталь с номером i изготавливается за p_i секунд. Специфика предприятия «Авто-2010» заключается в том, что там одновременно может изготавливаться лишь одна деталь двигателя. Для производства некоторых деталей необходимо иметь предварительно изготовленный набор других деталей.

Генеральный директор «Авто-2010» поставил перед предприятием амбициозную задачу — за наименьшее время изготовить деталь с номером 1, чтобы представить её на выставке.

Требуется написать программу, которая по заданным зависимостям порядка производства между деталями найдёт наименьшее время, за которое можно произвести деталь с номером 1.

Формат входных данных

Первая строка входного файла содержит число n ($1 \leqslant n \leqslant 100\,000$) — количество деталей двигателя.

Вторая строка содержит n натуральных чисел $p_1, p_2, \ldots, p_n$, определяющих время изготовления каждой детали в секундах. Время для изготовления каждой детали не превосходит 10^9 секунд.

Каждая из последующих n строк входного файла описывает характеристики производства деталей. Здесь i-я строка содержит число деталей k_i, которые требуются для производства детали с номером i, а также их номера. В i-й строке нет повторяющихся номеров деталей. Сумма всех чисел k_i не превосходит $200\,000$.

Известно, что не существует циклических зависимостей в производстве деталей.

Формат выходных данных

В первой строке выходного файла должны содержаться два числа: минимальное время (в секундах), необходимое для скорейшего производства детали с номером 1 и число k деталей, которые необходимо для этого произвести.

Во второй строке требуется вывести через пробел k чисел — номера деталей в том порядке, в котором следует их производить для скорейшего производства детали с номером 1.

Примеры

стандартный ввод	стандартный вывод
3 100 200 300 1 2 0 2 2 1	300 2 2 1
2 2 3 1 2 0	5 2 2 1
4 2 3 4 5 2 3 2 1 3 0 2 1 3	9 3 3 2 1

Задача I. Граф? А... А я думал, барон...

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

Однажды Маша прогуливалась по парку и нашла под деревом граф... Удивлены? Вы думали, что в этой задаче будет логичная обоснованная история? Не дождетесь! Так вот...

У Маши есть ориентированный граф, в i-й вершине которого записано некоторое целое положительное число a_i. Изначально Маша может положить в некоторую вершину графа монетку. За один ход разрешается переместить монетку, которая сейчас находится в некоторой вершине u, в любую вершину v, такую что в графе существует ориентированное ребро $u \to v$. Каждый раз, когда монетка оказывается в какой-либо вершине i, Маша записывает в блокнот число a_i (в частности, когда Маша изначально кладет монетку в некоторую вершину графа, она пишет в блокнот число, записанное в этой вершине). Маша хочет сделать ровно $k-1$ ходов таким образом, чтобы максимальное число, записанное ей в блокноте, было как можно меньше.

Формат входных данных

Первая строка содержит три целых числа n, m и k ($1 \leqslant n \leqslant 2 \cdot 10^5$, $0 \leqslant m \leqslant 2 \cdot 10^5$, $1 \leqslant k \leqslant 10^{18}$) — количество вершин и ребер в графе, а также количество ходов, которое хочет сделать Маша, соответственно.

Вторая строка содержит n целых чисел a_i ($1 \leqslant a_i \leqslant 10^9$) — числа, записанные в вершинах графа.

Каждая из следующих m строк содержит два целых числа u и v ($1 \leqslant u \neq v \leqslant n$) — это означает, что в графе существует ребро $u \to v$.

Гарантируется, что граф не содержит петель и кратных ребер.

Формат выходных данных

Выведите одно целое число — минимальное значение максимального числа, которое Маша выписала в блокнот, при оптимальном перемещении монетки.

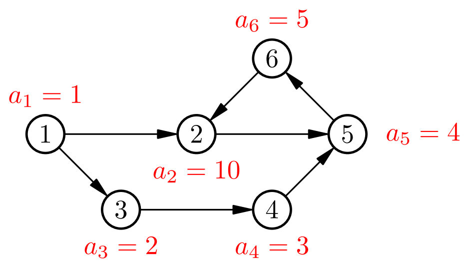
В случае, если Маше не удастся переместить монетку $k-1$ раз, в качестве ответа выведите число -1.

Примеры

стандартный ввод	стандартный вывод
6 7 4 1 10 2 3 4 5 1 2 1 3 3 4 4 5 5 6 6 2 2 5	4
6 7 100 1 10 2 3 4 5 1 2 1 3 3 4 4 5 5 6 6 2 2 5	10
2 1 5 1 1 1 2	-1
1 0 1 1000000000	1000000000

Замечание

На изображении ниже приведен граф, описанный в первых двух примерах.



В первом примере можно выбрать в качестве изначальной вершину 1. После этого можно выполнить три хода: $1 \to 3$, $3 \to 4$ и $4 \to 5$. В итоге в блокнот будут записаны числа $1, 2, 3$ и 4.

Во втором примере можно выбрать в качестве изначальной вершину 2. После этого можно выполнить 99 ходов: $2 \to 5$, $5 \to 6$, $6 \to 2$, $2 \to 5$, и так далее. В итоге в блокнот будут записаны числа $10, 4, 5, 10, 4, 5, \ldots, 10, 4, 5, 10$.

В третьем примере на имеющемся графе не удастся выполнить 4 хода.

Задача J. Кружки в Маховниках

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

Маленький Петя очень любит компьютеры и хочет научиться программировать.

В небольшом городке Маховники, где он живёт, работает сеть кружков по программированию самой разной тематики. Когда Петя пошёл записываться, он увидел большой список, состоящий из N кружков. Петя хочет быть всесторонне развитой личностью, поэтому он собрался отучиться во всех этих кружках. Но когда он собрался записаться на все занятия сразу, обнаружилось, что не всё так просто. Во-первых, в один момент времени разрешается учиться только в одном из этих N кружков. Во-вторых, некоторые преподаватели выдвигают входные требования к знаниям учеников, заключающиеся в знании курсов каких-то других кружков!

Петя хочет стать великим программистом, поэтому подобные мелочи его не останавливают. Действительно, ему достаточно всего-лишь составить правильный порядок посещения кружков, чтобы удовлетворить всем входным требованиям — это совсем простая задача, доступная даже совсем неопытному программисту.

Перед тем как сесть составлять порядок посещения кружков, Петя внимательно перечитал условия обучения и обнаружил ещё один важный пункт. Оказывается, для привлечения школьников, во всех кружках действует система поощрения учеников конфетами. Это означает, что по окончании очередного кружка ученику выдают несколько коробок конфет, всё больше и больше с каждым пройденным кружком. С другой стороны, в каждом кружке количество конфет в коробке своё, зависящее от сложности курса. Более конкретно — за прохождение i-го по счёту кружка, если этот кружок идёт в общем списке под номером j, ученику выдают аж $N^{i-1} \cdot j$ конфет — такие щедрые люди программисты.

Петя решил совместить полезное с приятным — теперь он хочет выбрать такой порядок посещения кружков, чтобы при этом получить как можно больше конфет, однако эта задача ему уже не под силу. Помогите будущему великому человеку отыскать такой порядок.

Формат входных данных

В первой строке входного файла содержится целое число N ($1 \leqslant N \leqslant 100000$) — количество кружков в Маховниках.

В последующих N строках идут описания входных требований кружков, в порядке их следования в общем списке. В i-й строке сначала записано целое число k_i ($0 \leqslant k_i \leqslant N-1$) — количество кружков, в которых нужно отучиться перед записью в i-й кружок, а потом k_i номеров этих кружков.

Сумма k_i не превосходит 200000.

Гарантируется, что возможно посетить все эти кружки в некотором порядке, не нарушая условия посещения.

Формат выходных данных

Выведите N номеров, разделённых пробелами — порядок, в котором Пете надо посещать кружки, чтобы съесть как можно больше конфет.

Пример

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

Задача K. Институт

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Недавно Антон закончил школу, и поступил в МФТИ. Однако, он быстро понял, что стоило поступить в нормальный вуз, а не в МФТИ, поэтому Антон решил потерять свой пропуск в университет, потому что если он это сделает, студент не сможет посещать занятия.

Кампус МФТИ можно представить как ориентированный граф. По некоторым ребрам графа можно перемещаться только при наличии пропуска в университет.

Для того, чтобы потерять пропуск, Антон должен выйти из общежития, которое расположено в вершине с номером 1, затем пройти по некоторому (возможно, нулевому) количеству ребер, затем оставить пропуск в некоторой вершине, после чего продолжить гулять по кампусу. В итоге у Антона не должно быть возможности вернуться в вершину, где он оставил пропуск.

Антону стало интересно, можно ли выполнить поставленную задачу: помогите ему определить, возможно ли потерять пропуск в кампусе таким образом, чтобы не было возможности вернуться в вершину, где был оставлен пропуск.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leqslant n, m \leqslant 300\,000$) — количество вершин и ребер в графе.

Каждая из следующих m строк содержит три целых числа u_i, v_i и t_i ($1 \leqslant u_i, v_i \leqslant n$, $1 \leqslant t_i \leqslant 2$) — описание ребра в графе. Ребро с номером i позволяет переместиться из вершины u_i в вершину v_i. Если для прохода по ребру необходим пропуск, $t_i = 1$. В противном случае $t_i = 2$.

Данный граф может содержать петли и кратные ребра.

Формат выходных данных

Если Антон может потерять пропуск таким образом, чтобы не было возможности вернуться в вершину, где он был оставлен, выведите «`Yes`». В противном случае выведите «`No`».

Примеры

стандартный ввод	стандартный вывод
3 4 1 2 1 2 3 2 3 2 1 3 1 2	Yes
6 8 1 2 1 2 3 2 3 2 2 3 4 1 4 1 2 1 5 2 5 4 2 6 1 2	No

Замечание

В первом примере Антон может пройти по ребру $1 \to 2$, воспользовавшись пропуском. Затем он может оставить пропуск в вершине 2 и пройти по ребру $2 \to 3$. После этого он не сможет вернуться в вершину 2.

Задача L. Траффик

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

Центр Гдынии расположен на острове посередине реки Кацза. Каждое утро тысячи машин едут из жилых районов на западном берегу в индустриальные на восточном.

Остров можно представить в виде прямоугольника $A \times B$ со сторонами, параллельными осям координат, углами $(0, 0)$ и (A, B).

На острове есть n перекрёстков, пронумерованных натуральными числами от 1 до n. Перекрёсток номер i имеет координаты (x_i, y_i). Если координаты какого-то перекрёстка имеют вид $(0, y)$, то он находится на западном берегу, если (A, y) — на восточном. Перекрёстки соединены дорогами, каждая — отрезок на плоскости, они не пересекаются (кроме концов). Дороги бывают односторонними и двусторонними. Нет никаких мостов и туннелей! Некоторые дороги могут идти по краям прямоугольника.

Поскольку плотность траффика растёт, мэр города нанял Вас (кого же ещё) проверить, достаточно ли текущей сети дорог. Он попросил написать программу, которая по карте города определит для каждого перекрёстка на западном берегу, сколько из него достижимых на восточном.

Формат входных данных

В первой строке входных данных дано четыре целых числа n, m, A и B ($1 \leqslant n \leqslant 300\,000$, $0 \leqslant m \leqslant 900\,000$, $1 \leqslant A \leqslant 10^9$, $1 \leqslant B \leqslant 10^9$). Это количества перекрёстков, дорог и размеры города, соответственно.

В каждой из следующих n строк есть два целых числа x_i и y_i ($0 \leqslant x_i \leqslant A$, $0 \leqslant y_i \leqslant B$), описывающих координаты перекрёстка номер i. Перекрёстки не совпадают.

Следующие m строк описывают дороги, каждая одну. Это описание состоит из трёх чисел: c_i — номер перекрёстка, откуда ($1 \leqslant c_i \leqslant n$), d_i — куда ($1 \leqslant d_i \leqslant n$), $k_i \in \{1, 2\}$ — тип дороги (сколько-сторонняя). Разные дороги соединяют разные неупорядоченные пары перекрёстков.

Есть хотя бы один перекрёстток на западном берегу, из которого можно добраться хотя бы в один на восточном по дорогам.

Формат выходных данных

Выведите для каждого перекрёстка на западном берегу в своей строчке количество достижимых из него перекрёстков на восточном берегу.

Выводите в порядке уменьшения y-координаты.

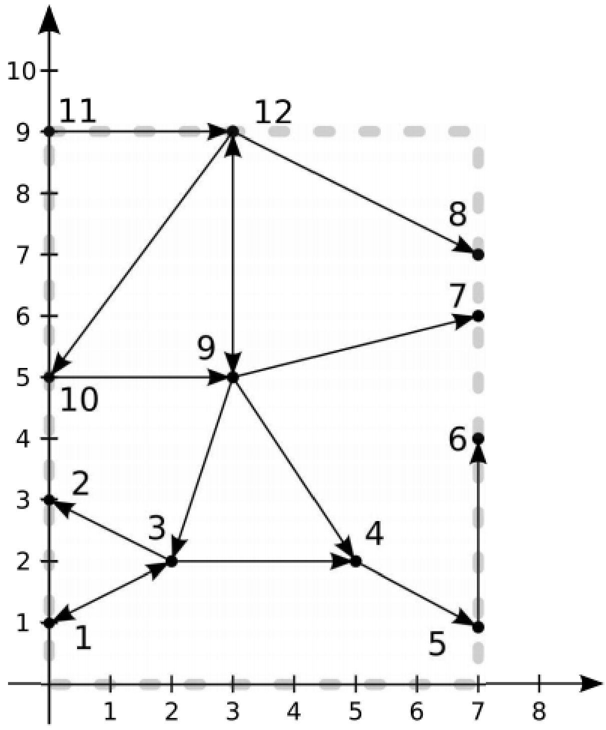
Система оценки

1. (30 баллов) $n, m \leqslant 6\,000$.
2. (70 баллов) $n \leqslant 300\,000$, $m \leqslant 900\,000$.

Примеры

стандартный ввод	стандартный вывод
5 3 1 3 0 0 0 1 0 2 1 0 1 1 1 4 1 1 5 2 3 5 2	2 0 2
12 13 7 9 0 1 0 3 2 2 5 2 7 1 7 4 7 6 7 7 3 5 0 5 0 9 3 9 1 3 2 3 2 1 3 4 1 4 5 1 5 6 1 9 3 1 9 4 1 9 7 1 9 12 2 10 9 1 11 12 1 12 8 1 12 10 1	4 4 0 2

Замечание


11
12
8
7
9
10
6
2
3
4
1
5
10
9
8
7
6
5
4
3
2
1
1
2
3
4
5
6
7
8

Задача M. Поздний контест

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

Много в мире разных часовых поясов! Именно поэтому соревнования по программированию часто бывают в неудобное для некоторых людей время. Как-то раз Гриша и Егор решили поучаствовать в соревновании, которое заканчивалось очень поздно. Именно поэтому ребята решили переночевать в гостинице, чтобы не возвращаться домой поздно. Однако все не так просто. Родители Егора и Гриши очень волнуются за своих детей, поэтому они решили установить по всему городу камеры, чтобы видеть, где находятся ребята.

В городе, где живут программисты $1 \leqslant n \leqslant 500\,000$ перекрестков, соединенных $n-1$ дорогой так, что между любыми двумя перекрестками существует путь по дорогам.

Родители собираются установить на перекрестках города камеры, радиус действия которых равен длине дороги. Родители будут спокойны, если смогут видеть ребят на любой дороге и на любом про перекрестке. Иными словами, для каждого перекрестка должен существовать перекресток, находящийся на расстоянии не более одной дороги, такой что на нем установлена камера и для любой дороги должна быть установлена камера хотя бы на одном конце этой дороги. Установка камер — затратное дело, поэтому для каждого перекрестка с номером i известна стоимость $1 \leqslant cost_i \leqslant 10^9$ установки камеры на нем.

Помогите родителям установить камеры надлежащим образом за минимальную стоимость.

Формат входных данных

В первой строке входного файла содержится количество перекрестков n $(1 \leqslant n \leqslant 500\,000)$.

В последующих $n-1$ строках содержатся v_i, u_i — перекрестки соединенные очередной дорогой.

Последняя строчка входного файла содержит n чисел $ost_1, ost_2, \ldots, ost_n$ $(1 \leqslant ost_i \leqslant 10^9)$ — стоимости установки камер на перекрестках.

Формат выходных данных

В первой строке выведете минимальную стоимость установки ans и количество перекрестков k, в которых надо установить камеры.

Во второй строке выведите k чисел — перекрестки, в которых надо установить камеры.

Если ответов несколько, разрешается вывести любой из них.

Пример

стандартный ввод	стандартный вывод
6 1 2 2 3 1 4 4 5 4 6 228 1488 2 2 8 1	232 3 1 3 4

Задача N. Два пути

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

Как вы знаете, Васин брат живет во Флатландии. Во Флатландии n городов, соединенных $n-1$ двухсторонней дорогой. Города пронумерованы от 1 до n. Из любого города можно добраться до любого другого.

Компания «Два пути», в которой работает Васин брат, выиграла тендер на ремонт двух путей во Флатландии. Путем называется последовательность различных городов, последовательно соединенных дорогами. Пути, которые надо отремонтировать, компания может выбрать самостоятельно. Единственное условие, накладываемое тендером, они не должны пересекаться (то есть иметь общих городов).

Известно, что прибыль, которую получит компания «Два пути», равна произведению длин выбранных двух путей. Считая длину каждой дороги один, а длину пути равной количеству дорог в ней, найдите максимальную возможную прибыль.

Формат входных данных

В первой строке записано целое число n ($2 \leqslant n \leqslant 100\,000$), где n — количество городов в стране.

Далее в $n-1$ строке записаны сами дороги. Каждая строка содержит пару номеров соединяемых дорогами a_i и b_i ($1 \leqslant a_i, b_i \leqslant n$).

Формат выходных данных

Выведите максимально возможную прибыль.

Примеры

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

Задача O. Ориентация

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Дан неориентированный граф, состоящий из N вершин и M ребер. Вершины пронумерованы целыми числами от 1 до N, а i-е ребро соединяет вершины a_i и b_i. Также дана последовательность из N целых чисел $c_1, c_2, \ldots, c_N$.

Вы должны заменить каждое неориентированное ребро (a_i, b_i) из графа на ориентированное ребро $a_i \to b_i$, либо на ориентированное ребро $b_i \to a_i$ таким образом, чтобы было выполнено следующее условие: для любого i от 1 до N существует ровно c_i вершин, достижимых из вершины i, включая саму вершину i.

Формат входных данных

Первая строка содержит два целых числа N и M ($1 \leqslant N \leqslant 100$, $0 \leqslant M \leqslant \frac{N(N-1)}{2}$).

Каждая из следующих M строк содержит два целых числа a_i и b_i ($1 \leqslant a_i, b_i \leqslant N$). Гарантируется, что граф не содержит петель и кратных ребер.

Последняя строка содержит N целых чисел $c_1, c_2, \ldots, c_N$ ($1 \leqslant c_i \leqslant N$).

Гарантируется, что для каждого теста существует решение.

Формат выходных данных

Выведите M строк. В i-й строке выведите «`->`» (без кавычек), если вместо i-го ребра нужно добавить в граф ориентированное ребро $a_i \to b_i$, либо «`<-`» (без кавычек) в противном случае.

Если существует несколько решений, выведите любое.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 2 3 3 1 3 3 3	-> -> ->
3 2 1 2 2 3 1 2 3	<- <-
6 3 1 2 4 3 5 6 1 2 1 2 2 1	<- -> ->

Замечание

В первом примере получился цикл длины 3. Из каждой вершины достижима каждая вершина.

Задача P. Сжатие таблицы

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

Петя увлёкся алгоритмами сжатия данных. Он уже изучил форматы *gz*, *bz*, *zip* и несколько других. Воодушевившись новыми знаниями, Петя собрался разработать свой формат сжатия и назвать его *dis*.

Петя решил сжимать таблицы. У него есть таблица, состоящая из n строк и m столбцов, заполненная целыми положительными числами. Он хочет заменить значения элементов таблицы на целые положительные числа так, чтобы отношение элементов в каждой строке и каждом столбце не изменилось. То есть, если в некоторой строке исходной таблицы $a_{i,j} < a_{i,k}$, то и в сжатой таблице $a'_{i,j} < a'_{i,k}$, и если $a_{i,j} = a_{i,k}$, то $a'_{i,j} = a'_{i,k}$. Аналогично, если в некотором столбце исходной таблицы $a_{i,j} < a_{p,j}$, то и в сжатой таблице $a'_{i,j} < a'_{p,j}$, и если $a_{i,j} = a_{p,j}$, то $a'_{i,j} = a'_{p,j}$.

Поскольку бо́льшие значения требуют больше места для хранения, максимальное значение элемента получившейся матрицы должно быть как можно меньше.

В теории Петя мастер, но вот писать код он не любит. Помогите ему реализовать формат сжатия *dis*.

Формат входных данных

В первой строке входных данных содержатся два числа n и m ($1 \leqslant n, m$ и $n \cdot m \leqslant 1\,000\,000$) — количество строк и столбцов таблицы соответственно.

В следующих n строках содержится по m целых чисел $a_{i,j}$ ($1 \leqslant a_{i,j} \leqslant 10^9$) — значения элементов таблицы.

Формат выходных данных

Выведите сжатую таблицу: n строк, содержащих по m чисел.

Если существует несколько ответов, минимизирующих максимальное число, то разрешается вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
2 2 1 2 3 4	1 2 2 3
4 3 20 10 30 50 40 30 50 60 70 90 80 70	2 1 3 5 4 3 5 6 7 9 8 7

Замечание

В первом примере $a_{1,2} \neq a_{2,1}$, но, поскольку они не располагаются в одной строке или в одном столбце, при сжатии их можно сделать равными.

Система оценки

Тесты к этой задаче состоят из восьми групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых предыдущих групп.

Группа	Тесты	Баллы	Дополнительные ограничения			Необх. группы	Комментарий
			n	m	$a_{i,j}$		
0	1 – 2	0	–			–	Тесты из условия
1	3 – 19	10	$n \leqslant 1000$	$m = 1$	–	–	
2	20 – 39	15	$n, m \leqslant 100$		Все $a_{i,j}$ различны	–	
3	40 – 74	15	$n, m \leqslant 100$		–	0, 2	
4	75 – 84	15	$n, m \leqslant 400$		Все $a_{i,j}$ различны	2	
5	85 – 102	15	$n, m \leqslant 400$		–	0, 2, 3, 4	
6	103 – 112	15	–		Все $a_{i,j}$ различны	2, 4	
7	–	15	–			0 – 6	

Задача Q. Операция «Перестановка»

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

Больше всего Генерал Петров любит строевую подготовку. Однажды в построении участвовало n солдат, для простоты пронумерованных от 1 до n. Солдаты выстроились в ряд, при этом на месте i стоял солдат с номером a_i. После этого адъютант генерала прошёл вдоль строя и проверил, что перестановка в точности совпадает с придуманной генералом.

Некоторое время спустя генерал захотел снова расставить солдат в том же порядке. Однако он забыл, как именно он расставил солдат в тот раз. К счастью, у адъютанта генерала хорошая память — про m пар позиций x_i, y_i он помнит, что номер солдата, стоявшего на месте x_i, меньше номера солдата, стоявшего на месте y_i.

Адъютант стал по очереди сообщать генералу пары x_i, y_i. Но генерал хочет побыстрее начать строить солдат. Помогите ему определить такое минимальное k, что как только адъютант сообщит ему первые k пар позиций, генерал сможет однозначно определить искомую перестановку.

Формат входных данных

В первой строке находятся два целых числа n и m — количество солдат, участвовавших в операции, и количество пар позиций, которые запомнил адъютант ($2 \leqslant n \leqslant 10^5$; $1 \leqslant m \leqslant 10^5$).

В следующих m строках даны пары, которые помнит адъютант, в том порядке, в котором сообщает их генералу. Каждая строка содержит по два числа x_i и y_i, которые означают, что номер солдата на позиции x_i был меньше номера солдата на позиции y_i ($1 \leqslant x_i, y_i \leqslant n$; $x_i \neq y_i$).

Гарантируется, что каждая пара x_i, y_i встречается во входном файле не более одного раза. Гарантируется, что входные данные корректные, то есть существует хотя бы одна перестановка, для которой выполняются все условия, которые помнит адъютант.

Формат выходных данных

Выведите одно число k — минимальный номер пары позиций, после произнесения которой можно однозначно восстановить перестановку. Если входные данные таковы, что перестановку однозначно восстановить не удастся, выведите -1.

Примеры

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

Замечание

В первом примере солдаты стояли в порядке $(5, 2, 1, 3, 4)$. Уже после четвёртой пары чисел, запомненной адъютантом, можно восстановить этот порядок.

Во втором примере существует четыре варианта расстановки солдат, удовлетворяющих входным данным: $(1, 3, 4, 2)$, $(2, 3, 4, 1)$, $(3, 2, 4, 1)$ и $(4, 2, 3, 1)$.

Задача R. История о кротах

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	1024 мегабайта

Крот — маленький тщедушный зверек...

Примечание переводчика

Не так давно семья кротов решила прокопать новую сеть туннелей под землей. План сети, который уже разработан, состоит из подземных комнат и туннелей, которые их соединяют. Формально, план сети представляет собой неориентированный связный граф. Кротиха-мать хочет использовать данную работу как возможность научить ее двоих детей-кротов, как правильно вырыть сеть туннелей.

В качестве начальной быстрой демонстрации кротиха-мать собирается вырыть несколько комнат и туннелей, которые образуют простой путь в графе плана сети. После этого она разделит оставшиеся комнаты между своими двумя детьми так, чтобы каждому из детей досталось равное количество комнат, иначе они могут расстроиться (насчет туннелей можно не беспокоиться, потому что их копать гораздо проще). Дети могут работать в любых из своих комнат в любом порядке, в котором они захотят.

Так как у детей нет опыта по копанию сети туннелей, кротиха-мать поняла одну проблему: если между некоторой парой комнат, которые принадлежат разным детям, в плане есть туннель, то существует риск происшествия во время выкапывания данного туннеля.

Помогите кротихе-матери решить, какой путь использовать для начальной демонстрации, и как разделить оставшиеся комнаты поровну так, чтобы не существовало пары комнат, отданных разным детям, соединенных туннелем. Изначальный путь должен содержать хотя бы одну комнату и не должен посещать какую-либо комнату два раза.

Формат входных данных

В первой строке записаны два целых числа c и t ($1 \leqslant c \leqslant 2 \cdot 10^5$, $0 \leqslant t \leqslant 2 \cdot 10^5$) — количество комнат и туннелей в плане.

В каждой из следующих t строк записаны два целых числа a и b ($1 \leqslant a, b \leqslant c$, $a \neq b$) — номера комнат, соединенных туннелем.

Комнаты пронумерованы числами от 1 до c. Между каждой парой комнат есть не более одного туннеля. Между любой парой комнат существует путь.

Формат выходных данных

В первой строке выведите два целых числа p и s — количество комнат в пути, выбранном кротихой-матерью для демонстрации, и количество комнат, которые должен быть вырыть каждый из детей.

Во второй строке выведите p чисел — номера комнат, образующих путь для демонстрации. Комнаты должны быть выведены в порядке следования по пути.

В третьей строке выведите s чисел — номера комнат, отданных первому ребенку в произвольном порядке.

В четвертой строке выведите s чисел — номера комнат, отданных второму ребенку в произвольном порядке.

Гарантируется, что ответ существует. Если существует несколько корректных ответов, выведите любой.

Примеры

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

Задача S. Редукция дерева

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Задано неориентированное дерево, содержащее n вершин. Можно выбрать некоторое ребро и удалить его, при этом инцидентные ему вершины не удаляются. Таким образом можно удалить из дерева некоторый набор рёбер. В результате дерево распадается на некоторое количество меньших деревьев. Требуется, удалив наименьшее количество рёбер, получить в качестве хотя бы одной из компонент связности дерево, содержащее ровно p вершин.

Формат входных данных

Первая строка входного файла содержит пару натуральных чисел n и p ($1 \leqslant p \leqslant n \leqslant 5000$). Далее в $n-1$ строке содержатся описания рёбер дерева. Каждое описание состоит из пары натуральных чисел a_i, b_i ($1 \leqslant a_i, b_i \leqslant n$) — номеров соединяемых ребром вершин.

Формат выходных данных

В первую строку выведите наименьшее количество рёбер q в искомом наборе. Во вторую строку выведите номера удаляемых рёбер. Номера рёбер определяются порядком их задания по входному файлу. Рёбра нумеруются с единицы. Если оптимальных решений несколько, разрешается выводить любое.

Пример

стандартный ввод	стандартный вывод
11 6 1 2 1 3 1 4 2 6 2 7 1 5 2 8 4 9 4 10 4 11	2 3 6