

## Задача А. Поиск цикла

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Дан ориентированный невзвешенный граф. Необходимо определить есть ли в нём циклы, и если есть, то вывести любой из них.

### Формат входных данных

В первой строке входного файла находятся два натуральных числа  $N$  и  $M$  ( $1 \leq N \leq 100\,000$ ,  $M \leq 100\,000$ ) — количество 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

Если в графе нет цикла, то вывести «NO», иначе — «YES» и затем перечислить все вершины в порядке обхода цикла.

### Примеры

stdin	stdout
2 2 1 2 2 1	YES 2 1
2 2 1 2 1 2	NO

## Задача В. Построение

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

Группа солдат-новобранцев прибыла в армейскую часть №666. После знакомства с прапорщиком стало очевидно, что от работ на кухне по очистке картофеля спасти солдат может только чудо.

Прапорщик, будучи не в состоянии запомнить фамилии, пронумеровал новобранцев от 1 до  $N$ . После этого он велел им построиться по росту (начиная с самого высокого). С этой несложной задачей могут справиться даже совсем необученные новобранцы, да вот беда, прапорщик уверил себя, что знает про некоторых солдат, кто из них кого выше, и это далеко не всегда соответствует истине.

После трех дней обучения новобранцам удалось выяснить, что знает (а точнее, думает, что знает) прапорщик. Помогите им, используя эти знания, построиться так, чтобы товарищ прапорщик остался доволен.

### Формат входных данных

Сначала на вход программы поступают числа  $N$  и  $M$  ( $2 \leq N \leq 10^5$ ,  $1 \leq M \leq 2 \cdot 10^5$ ) — количество солдат в роте и количество пар солдат, про которых прапорщик знает, кто из них выше.

Далее идут эти пары чисел  $A$  и  $B$  по одной на строке ( $1 \leq A, B \leq N$ ), что означает, что, по мнению прапорщика, солдат  $A$  выше, чем  $B$ .

Не гарантируется, что все пары чисел во входных данных различны.

### Формат выходных данных

В первой строке выведите «Yes» (если можно построиться так, чтобы прапорщик остался доволен) или «No» (если нет).

После ответа «Yes» на следующей строке выведите  $N$  чисел, разделенных пробелами, — одно из возможных построений.

### Примеры

стандартный ввод	стандартный вывод
2 2 1 2 2 1	No
3 7 1 2 2 3 1 3 2 3 1 2 1 2 1 3	Yes 1 2 3

## Задача С. 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 D. 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 Е. 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  $n$  ( $1 \leq n \leq 10^5$ ) — количество деталей двигателя.

Вторая строка содержит  $n$  натуральных чисел  $p_1, p_2, \dots, p_n$ , 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  $2 \cdot 10^5$ .

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 F. Наводнение

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

На базе «Берендеевы поляны» все домики соединены между собой дорожками. Каждая дорожка имеет свою высоту над уровнем озера Лель. Каждый домик может быть соединён с другим несколькими дорожками, но дорожек, ведущих из домика в него же, нет.

После сильных дождей озеро выходит из берегов и затапливает дорожки. Дорожка затапливается в том случае, если уровень воды больше или равен её высоте. Поскольку в лагере есть только самокаты и велосипеды, а лодок ещё не закупили, для нормального функционирования лагеря от любого домика до любого другого домика должен существовать путь по незатопленным дорожкам, возможно, по нескольким. Помогите сотрудникам базы определить минимальный уровень воды, при котором найдутся такие два домика, что от одного из них никак нельзя добраться до другого. Гарантируется, что до начала сильных дождей из любого домика можно было попасть в любой другой.

### Формат входных данных

В первой строке даны два числа  $A$  и  $B$  — количество домиков и дорожек соответственно ( $2 \leq A \leq 10\,000$ ,  $1 \leq B \leq 20\,000$ ). Следующие  $B$  строк содержат тройки чисел  $(s_i, f_i, h_i)$ , где  $s_i$  и  $f_i$  — номера домиков, которые соединяет  $i$ -я дорожка, а  $h_i$  — её высота над уровнем озера в миллиметрах ( $1 \leq s_i \leq A$ ,  $1 \leq f_i \leq A$ ,  $1 \leq h_i \leq 1\,000\,000$ ).

### Формат выходных данных

Выведите минимальную высоту, на которую должна подняться вода, чтобы в лагере нашлось хотя бы два домика, между которыми нельзя пройти по незатопленным дорожкам.

### Примеры

стандартный ввод	стандартный вывод
2 1 1 2 100	100
4 5 1 2 100 1 3 400 2 3 300 2 4 200 3 4 500	300

## Задача G. Премьер министр

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

Новый премьер-министр решил проехать по России от Москвы до Владивостока по железной дороге, а затем вернуться обратно. Он поручил своим помощникам разработать маршрут так, чтобы не пришлось два раза проезжать через один и тот же город. Однако помощники сообщили, что для Российских железных дорог это невозможно. Определите, в каких городах премьер-министр будет вынужден побывать дважды.

### Формат входных данных

В первой строке входных данных находятся числа  $n$  и  $m$  — количество российских городов, соединенных железными дорогами в единую сеть, и количество железнодорожных перегонов, соединяющих пары городов ( $1 \leq n \leq 20\,000$ ;  $1 \leq m \leq 200\,000$ ).

Города имеют номера от 1 до  $n$ . В каждой из следующих  $m$  строк находится пара натуральных чисел, описывающая, между какими двумя городами проходит соответствующая железнодорожная ветка. В последней строке находятся два целых числа  $s$  и  $e$  ( $1 \leq s \neq e \leq n$ ) — номера Москвы и Владивостока по версии РЖД.

### Формат выходных данных

В первой строке выведите число  $b$  — количество городов, которые премьер-министру придется посетить дважды. В следующих  $b$  строках выведите  $b$  целых чисел — номера этих городов в возрастающем порядке.

### Примеры

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

## Задача Н. 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 воздушные 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