

## Задача А. Эйлеров путь

Имя входного файла: `euler.in`  
Имя выходного файла: `euler.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дан неориентированный связный граф, не более трех вершин имеет нечетную степень. Требуется определить, существует ли в нем путь, проходящий по всем ребрам.

Если такой путь существует, необходимо его вывести.

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  — количество вершин графа ( $1 \leq n \leq 100\,000$ ). Далее следуют  $n$  строк, задающих ребра. В  $i$ -й из этих строк находится число  $m_i$  — количество ребер, инцидентных вершине  $i$ . Далее следуют  $m_i$  натуральных чисел — номера вершин, в которые ведут ребро из  $i$ -й вершины.

Граф может содержать кратные ребра, но не содержит петель.

Граф содержит не более 300 000 ребер.

### Формат выходных данных

Если решение существует, то в первую строку выходного файла выведите одно число  $k$  — количество ребер в искомом маршруте, а во вторую  $k + 1$  число — номера вершин в порядке их посещения.

Если решений нет, выведите в выходной файл одно число `-1`.

Если решений несколько, выведите любое.

### Пример

<code>euler.in</code>	<code>euler.out</code>
4	5
2 2 2	1 2 3 4 2 1
4 1 4 3 1	
2 2 4	
2 3 2	

## Задача В. 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 ( $1 \leq n \leq 200\,000$ ,  $1 \leq m \leq 200\,000$ ).

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

### Формат выходных данных

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

### Пример

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

---

## Задача С. Ленивый Глеб

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

Недавно Глеба попросили придумать задачу на одну из олимпиад. Так ничего и не придумав, он обратился к своему давнему другу Владу, у которого за спиной огромное количество разработанных задач. Влад очень любит деревья и запросы на них. У Влада есть дерево из  $n$  вершин, занумерованных от 1 до  $n$ . В каждой вершине дерева изначально стоит какое-то число  $a_i$ ,  $1 \leq a_i \leq 1e9$ . Также существуют два вида запросов :

- 1) Запрос вида  $a, b$  — требуется найти  $\oplus$  на пути от  $a$  до  $b$
- 2) Запрос вида  $x$  — требуется все значения  $a_i$  сделать равными  $a_i \oplus x$

Глебу очень понравилась эта задача и поэтому он решил сделать ее, но к сожалению у него начался турнир по одной классной игре и ему стало лень делать тесты, в итоге он решил, что во всех вершинах дерева изначально стоят нули.

### Формат входных данных

На вход даются два числа  $n$  — количество вершин в дереве и  $m$  — количество запросов к дереву. На второй строке —  $n - 1$  число, где  $i$ -е число — индекс  $p$  — предка вершины  $1 \leq p \leq i - 1$ ,  $n, m \leq 10^6$ . Затем в следующих  $m$  строках даются запросы, сначала дается  $t$  - тип запроса, а затем либо  $a, b$  - если требуется найти  $\oplus$  на пути, либо  $x$ , если запрос второго типа,  $x \leq 10^9$

### Формат выходных данных

Требуется для каждого запроса первого типа вывести  $\oplus$  на пути

## Задача D. Точки сочленения

Имя входного файла: `points.in`  
Имя выходного файла: `points.out`  
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 ( $1 \leq n \leq 20\,000$ ,  $1 \leq m \leq 200\,000$ ).

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.

### Пример

<code>points.in</code>	<code>points.out</code>
6 7	2
1 2	2 3
2 3	
2 4	
2 5	
4 5	
1 3	
3 6	

## Задача Е. 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.

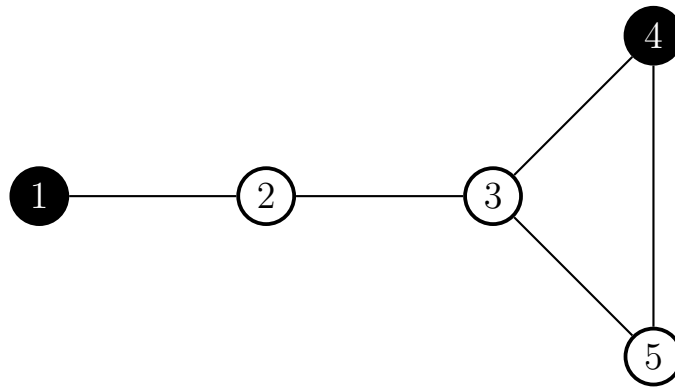


Рис. 1: Пример сети и отказоустойчивого множества серверов.

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,  $s$  — 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 ( $2 \leq n \leq 200\,000$ ,  $1 \leq m \leq 200\,000$ ). 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	2 3
1 2	
2 3	
3 4	
3 5	
4 5	

## Задача F. Мосты

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

На курсах Ginkoff Teneration паника. Через неделю все школьники, не решившие 20 задач, вылетят. Но слава богу, вы не из таких, вы даже готовы послушать лекцию про мосты. Более того, вы достаточно смелы, чтобы пробраться через условия других жестких задач и домотать до самого низа pdf-ки с условиями. За это вы вознаграждаетесь самой простой задачей конкурса.

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.

### Пример

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