

## Задача А. Компоненты реберной двусвязности

Имя входного файла: `bicone.in`  
Имя выходного файла: `bicone.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Компонентой реберной двусвязности графа  $\langle V, E \rangle$  называется подмножество вершин  $S \subset V$ , такое что для любых различных  $u$  и  $v$  из этого множества существует не менее двух реберно не пересекающихся путей из  $u$  в  $v$ .

Дан неориентированный граф. Требуется выделить компоненты реберной двусвязности в нем.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и ребер графа соответственно ( $1 \leq n \leq 20\,000$ ,  $1 \leq m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание ребер 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

В первой строке выходного файла выведите целое число  $k$  — количество компонент реберной двусвязности графа. Во второй строке выведите  $n$  натуральных чисел  $a_1, a_2, \dots, a_n$ , не превосходящих  $k$ , где  $a_i$  — номер компоненты реберной двусвязности, которой принадлежит  $i$ -я вершина.

### Примеры

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

## Задача В. Компоненты вершинной двусвязности

Имя входного файла: `biconv.in`  
Имя выходного файла: `biconv.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Компонентой вершинной двусвязности графа  $\langle V, E \rangle$  называется максимальный по включению подграф (состоящий из вершин и ребер), такой что любые два ребра из него лежат на вершинно простом цикле.

Дан неориентированный граф без петель. Требуется выделить компоненты вершинной двусвязности в нем.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и ребер графа соответственно ( $1 \leq n \leq 20\,000$ ,  $1 \leq m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание ребер 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

В первой строке выходного файла выведите целое число  $k$  — количество компонент вершинной двусвязности графа. Во второй строке выведите  $m$  натуральных чисел  $a_1, a_2, \dots, a_m$ , не превосходящих  $k$ , где  $a_i$  — номер компоненты вершинной двусвязности, которой принадлежит  $i$ -е ребро. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

### Примеры

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

## Задача С. Компьютерная сеть

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Компьютерная сеть «Plunder & Flee Inc.» состоит из  $n$  серверов и  $m$  двусторонних каналов связи. Два сервера могут общаться либо по прямому каналу, либо по цепочке каналов, передавая информацию от сервера к серверу. Текущая настройка сети обеспечивает связь для любой пары серверов.

Сетевой администратор стремится максимально повысить надежность сети. Некоторые каналы связи в сети были определены как критические. Сбой на любом критическом канале разделит сеть на несвязанные сегменты. Руководство компании отреагировало на опасения администратора и согласилось профинансировать ещё один канал связи, при условии, что когда новый канал появится в сети, количество критических каналов будет сведено к минимуму.

Напишите программу, которая по заданной конфигурации сети будет выбирать пару серверов подключиться по новой линии связи. Если несколько таких пар позволяют минимизировать количество критических ссылок, то любая из них будет считаться правильным ответом.

### Формат входных данных

В первой строке содержится два натуральных числа  $n$  и  $m$  ( $1 \leq n \leq 10\,000$ ,  $1 \leq m \leq 100\,000$ ). В следующих  $m$  строках через пробел заданы по два натуральных числа  $x_i$  и  $y_i$ , которые описывают номера серверов соединённых каналом ( $1 \leq x_i, y_i \leq n$ ,  $x_i \neq y_i$ ).

### Формат выходных данных

Выведите пару чисел через пробел, номера серверов, которые необходимо соединить каналом, чтобы минимизировать количество критических каналов.

### Примеры

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

## Задача D. Почтальон

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

В городе  $\mathcal{C}$  есть  $n$  площадей, соединенных улицами. При этом количество улиц не превышает ста тысяч и существует не более трех площадей, на которые выходит нечетное число улиц. Для каждой улицы известна ее длина. По улицам разрешено движение в обе стороны. В городе есть хотя бы одна улица. От любой площади до любой можно прийти по улицам.

Почтальону требуется пройти хотя бы один раз по каждой улице так, чтобы длина его пути была наименьшей. Он может начать движение на любой площади и закончить также на любой (в том числе и на начальной).

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  — количество площадей в городе ( $1 \leq n \leq 1000$ ). Далее следуют  $n$  строк, задающих улицы. В  $i$ -й из этих строк находится число  $m_i$  — количество улиц, выходящих из площади  $i$ . Далее следуют  $m_i$  пар положительных чисел. В  $j$ -й паре первое число — номер площади, в которую идет  $j$ -я улица с  $i$ -й площади, а второе число — длина этой улицы.

Между двумя площадями может быть несколько улиц, но не может быть улиц с площади на нее саму.

Все числа во входном файле не превосходят  $10^5$ , общее количество улиц также не превосходит  $10^5$ .

### Формат выходных данных

Если решение существует, то в первую строку выходного файла выведите одно число — количество улиц в искомом маршруте (считая первую и последнюю), а во вторую — номера площадей в порядке их посещения.

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

Если решений несколько, выведите любое.

### Примеры

<code>post.in</code>	<code>post.out</code>
4	5
2 2 1 2 2	1 2 4 3 2 1
4 1 2 4 4 3 5 1 1	
2 2 5 4 8	
2 3 8 2 4	

## Задача Е. Таня и пароль

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

Пока папа был на работе, маленькая девочка Таня решила поиграть с папиным паролем к секретной базе данных. Папин пароль представляет собой строку, состоящую из  $n + 2$  символов. Она выписала все возможные  $n$  трёхбуквенных подстрок пароля на бумажки, по одной на каждую бумажку, а сам пароль выкинула. Каждая трёхбуквенная подстрока была выписана на бумажки столько раз, сколько она встречалась в пароле. Таким образом, в итоге у Тани оказалось  $n$  бумажек.

Потом Таня поняла, что папа расстроится, если узнает о ее игре, и решила восстановить пароль или, по крайней мере, хотя бы какую-то строку, соответствующую получившемуся набору трёхбуквенных строк. Вам предстоит помочь ей в этой непростой задаче. Известно, что папин пароль состоял из строчных и заглавных букв латинского алфавита, а также из цифр. Строчные и заглавные буквы латинского алфавита считаются различными.

### Формат входных данных

В первой строке следует целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ), количество трёхбуквенных подстрок, которые получились у Тани.

Следующие  $n$  строк каждая содержат по три буквы, образующие подстроку пароля папы. Каждый символ во вводе — строчная или заглавная буква латинского алфавита или цифра.

### Формат выходных данных

Если во время игры Таня что-то напутала, и строк, соответствующих данному набору подстрок, не существует, то выведите «NO».

Если же возможно восстановить строку, соответствующую данному набору подстрок, то выведите «YES», а затем любой подходящий вариант пароля.

### Примеры

стандартный ввод	стандартный вывод
5 aca aba aba cab bac	YES abacaba
4 abc bCb cb1 b13	NO
7 aaa aaa aaa aaa aaa aaa aaa	YES aaaaaaaaa