

Задача А. Компоненты связности (1 балл)

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

Дан неориентированный граф. Требуется выделить компоненты связности в нем.

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leq b_i, e_i \leq n$). Допускаются петли и параллельные ребра.

Формат выходного файла

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

Пример

<code>components.in</code>	<code>components.out</code>
3 1 1 2	2 1 1 2
4 2 1 3 2 4	2 2 1 2 1

Задача В. Мосты (1 балл)

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

Дан неориентированный граф. Требуется найти все мосты в нем.

Формат входного файла

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

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 выходного файла

Первая строка выходного файла должна содержать одно натуральное число b — количество мостов в заданном графе. На следующей строке выведите b целых чисел — номера ребер, которые являются мостами, в возрастающем порядке. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Пример

bridges.in	bridges.out
6 7	1
1 2	3
2 3	
3 4	
1 3	
4 5	
4 6	
5 6	

Задача С. Точки сочленения (1 балл)

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

Дан неориентированный граф. Требуется найти все точки сочленения в нем.

Формат входного файла

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

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 выходного файла

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>
9 12	3
1 2 1 3	1
2 3 1 4	2
4 5 1 5	3
2 6 6 7	
2 7 3 8	
8 9 3 9	

Задача D. Двудольный граф (1 балл)

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

Двудольным называется неориентированный граф $\langle V, E \rangle$, вершины которого можно разбить на два множества L и R , так что $L \cap R = \emptyset$, $L \cup R = V$ и для любого ребра $(u, v) \in E$ либо $u \in L, v \in R$, либо $v \in L, u \in R$.

Дан неориентированный граф. Требуется проверить, является ли он двудольным.

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leq b_i, e_i \leq n$). Допускаются петли и параллельные ребра.

Формат выходного файла

В единственной строке выходного файла выведите «YES», если граф является двудольным и «NO» в противном случае.

Пример

bipartite.in	bipartite.out
4 4 1 2 1 3 2 4 4 2	YES
3 3 1 2 2 3 3 1	NO

Задача Е. Компоненты реберной двусвязности (2 балла)

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 пути из u в v .

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

Формат входного файла

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

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 выходного файла

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

Пример

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

Задача F. Компоненты вершинной двусвязности (2 балла)

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 подмножество ребер $S \subset E$, такое что любые два ребра из него лежат на вершинно простом цикле.

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

Формат входного файла

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

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 выходного файла

В первой строке выходного файла выведите целое число k — количество компонент вершинной двусвязности графа. Во второй строке выведите m натуральных чисел a_1, a_1, \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.

Пример

<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	