

## Задача А. 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 ( $1 \leq n, m \leq 10^5$ ). 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 В. Topsort

Имя входного файла: `topsort.in`  
Имя выходного файла: `topsort.out`  
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  $M$  ( $1 \leq N \leq 100\,000, 0 \leq M \leq 100\,000$ ) — 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

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

## Задача С. Конденсация графа. Light.

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

Требуется найти количество рёбер в конденсации ориентированного графа. Примечание: конденсация графа не содержит кратных рёбер и петель.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и рёбер графа соответственно ( $n \leq 10\,000, m \leq 100\,000$ ). Следующие  $m$  строк содержат описание рёбер, по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — началом и концом ребра соответственно ( $1 \leq b_i, e_i \leq n$ ). В графе могут присутствовать кратные рёбра и петли.

### Формат выходных данных

Первая строка выходного файла должна содержать одно число — количество рёбер в конденсации графа.

### Пример

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

## Задача D. 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 ( $1 \leq N \leq 200\,000$ ,  $1 \leq M \leq 200\,000$ ). 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	2
1 4	1 2 2 1 1 2 2 2 2 1
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	

## Задача E. 2-SAT

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

Формулировка 2-SAT: нужно подобрать значения  $n$  булевых переменных так, чтобы все  $m$  утверждений вида  $x_{i_1} = e_1 \vee x_{i_2} = e_2$  обратились в истину. В данной задаче вам гарантируется, что решение существует.

### Формат входных данных

Входной файл состоит из одного или нескольких тестов.

Каждый тест описывается следующим образом. На первой строке число переменных  $n$  и число утверждений  $m$ . Каждая из следующих  $m$  строк содержит числа  $i_1, e_1, i_2, e_2$ , задает утверждение  $x_{i_1} = e_1 \vee x_{i_2} = e_2$  ( $0 \leq i_j < n$ ,  $0 \leq e_j \leq 1$ ). Ограничения: сумма всех  $n$  не больше 100 000, сумма всех  $m$  не больше 300 000.

### Формат выходных данных

Для каждого теста выведите строку из  $n$  нулей и единиц — значения переменных. Если у данной задачи 2-SAT есть несколько решений, выведите любое.

### Пример

стандартный ввод	стандартный вывод
1 0	0
2 2	01
0 0 1 0	000
0 1 1 1	
3 4	
0 1 1 0	
0 0 2 1	
1 1 2 0	
0 0 0 1	