

А. Компоненты связности

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Дан неориентированный граф. Требуется выделить компоненты связности в нем.

Входные данные

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \leq n \leq 50\,000$, $0 \leq m \leq 100\,000$).

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.

Выходные данные

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.

В. Поиск цикла

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

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

Входные данные

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.

Выходные данные

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

С. Топологическая сортировка

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Входные данные

В первой строке входного файла даны два натуральных числа N и M ($1 \leq N \leq 50\,000$, $0 \leq M \leq 10\,000$) — 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.

Выходные данные

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести —1.

D. Конденсация графа

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Конденсацией графа G называется новый граф H , где каждой компоненте сильной связности в графе G соответствует вершина из графа H . Ребро vu в графе H есть тогда и только тогда, когда в графе G существует хотя бы одно ребро из соответствующей v компоненте сильной связности, в компоненту, соответствующую u .

Требуется найти количество ребер в конденсации ориентированного графа.

Примечание: конденсация графа не содержит кратных ребер.

Входные данные

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($n \leq 10\,000$, $m \leq 100\,000$). Следующие m строк содержат описание ребер, 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 ребра и петли.

Выходные данные

Единственная строка выходного файла должна содержать одно число — количество ребер в конденсации графа.