Лабораторная работа 2-1. Графы, DFS, MST

A. 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 ($1 \le N \le 100\,000$, $0 \le M \le 100\,000$) — 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

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

Пример

входные данные	Скопировать
6 6 1 2 3 2 4 2 2 5 6 5 4 6	

выходные данные	Скопировать
4 6 3 1 2 5	

B. Мосты

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

Дан неориентированный граф, не обязательно связный, но не содержащий петель и кратных рёбер. Требуется найти все мосты в нём.

Входные данные

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

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \le b_i, e_i \le n$).

Выходные данные

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

входные данные	Скопировать
6 7 1 2 2 3 3 4 1 3 4 5 4 6 5 6	

выходные данные	Скопировать
1 3	

C. Точки сочленения

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

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

Входные данные

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

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \le b_i, e_i \le n$).

Выходные данные

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

входные данные	Скопировать
6 7 1 2 2 3 2 4 2 5 4 5 1 3 3 6	

выходные данные	Скопировать
2 2 3	

D. Компоненты реберной двусвязности

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 64 мегабайта
ввод: стандартный ввод
вывод: стандартный вывод

Компонентой реберной двусвязности графа называется подмножество вершин , 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.

Входные данные

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

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

Выходные данные

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

Пример

входные данные	Скопировать
6 7 1 2 2 3 3 1 1 4 4 5 4 6 5 6	

выходные данные	Скопировать
2 1 1 1 2 2 2	

E. Компоненты вершинной двусвязности

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 64 мегабайта
ввод: стандартный ввод
вывод: стандартный вывод

Компонентой вершинной двусвязности графа 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.

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

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

Выходные данные
В первой строке выходного файла выведите целое число k — количество компонент вершинной двусвязности графа. Во второй строке выведите m натуральных чисел $a_1, a_2, \ldots, a_m$, 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

входные данные	Скопировать
5 6 1 2 2 3 3 1 1 4 4 5 5 1	

выходные данные	Скопировать
2 1 1 1 2 2 2	

F. Конденсация графа

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

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 ($n \le 10\,000$, $m \le 100\,000$). 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 ($1 \le b_i, e_i \le n$). В графе могут присутствовать кратные ребра и петли.

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

Пример

входные данные	Скопировать
4 4 2 1 3 2 2 3 4 3	

выходные данные	Скопировать
2	

G. Планирование вечеринки

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Петя планирует вечеринку, это дело непростое. Одна из главных проблем в том, что некоторые его друзья плохо ладят друг с другом, а некоторые — наоборот. В результате у него есть множество требований, например: «Я приду только если придет Гена» или «Если там будет

Марина, то меня там точно не будет».

Петя формализовал все требования в следующем виде: «[+-]name1 => [+-]name2», здесь «name1» и «name2» — имена двух друзей Пети, «+» означает, что друг придет в гости, «-» — что не придет. Например, выражение «Если Андрея не будет, то Даша не придет» записывается так: «-andrey => -dasha».

Помогите Пете составить хоть какой-нибудь список гостей, удовлетворяющий всем свойствам, или скажите, что это невозможно

Входные данные
В первой строке входного файла записаны числа n и m — число друзей Пети и число условий ($1 \le n, m \le 1000$). В следующих n строках записаны имена друзей. Имена друзей состоят из маленьких латинских букв и имеют длину не больше 10. В следующих m строках записаны условия.

Выходные данные
Выведите в первой строке число k — число друзей, которых нужно пригласить. В следующих k строках выведите их имена.

Примеры

входные данные	Скопировать
3 3 vova masha gosha -vova => -masha -masha => +gosha +gosha => +vova	

выходные данные	Скопировать
2 vova masha	

входные данные	Скопировать
1 1 vova -vova => +vova	

выходные данные	Скопировать
1 vova	

входные данные	Скопировать
2 4 vova masha +vova => +masha +masha => -vova -vova => -masha -masha => +vova	

выходные данные	Скопировать
-1	

H. Авиаперелеты

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: avia.in
вывод: avia.out

Главного конструктора Петю попросили разработать новую модель самолета для компании «Air Бубундия». Оказалось, что самая сложная часть заключается в подборе оптимального размера топливного бака.

Главный картограф «Air Бубундия» Вася составил подробную карту Бубундии. На этой карте он отметил расход топлива для перелета между каждой парой городов.

Петя хочет сделать размер бака минимально возможным, для которого самолет сможет долететь от любого города в любой другой (возможно, с дозаправками в пути).

Входные данные
Первая строка входного файла содержит натуральное число n ($1 \le n \le 1000$) — число городов в Бубундии.

Далее идут n строк по n чисел каждая. j-ое число в i-ой строке равно расходу топлива при перелете из i-ого города в j-ый. Все числа не меньше нуля и меньше 10^9. Гарантируется, что для любого i в i-ой строчке i-ое число равно нулю.

Выходные данные
Первая строка выходного файла должна содержать одно число — оптимальный размер бака.

Пример

входные данные	Скопировать
4 0 10 12 16 11 0 8 9 10 13 0 22 13 10 17 0	

выходные данные	Скопировать
10	

I. Остовное дерево

ограничение по времени на тест: 4 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Даны точки на плоскости, являющиеся вершинами полного графа. Вес ребра равен расстоянию между точками, соответствующими концам этого ребра. Требуется в этом графе найти остовное дерево минимального веса.

Входные данные

Первая строка входного файла содержит натуральное число n — количество вершин графа ($1 \le n \le 10\,000$). Каждая из следующих n строк содержит два целых числа x_i, y_i — координаты i-й вершины ($-10\,000 \le x_i, y_i \le 10\,000$). Никакие две точки не совпадают.

Выходные данные

Первая строка выходного файла должна содержать одно вещественное число — вес минимального остовного дерева.

Пример

входные данные	Скопировать
2 0 0 1 1	

выходные данные	Скопировать
1.4142135624	

J. Остовное дерево 2

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

Требуется найти в связном графе остовное дерево минимального веса.

Входные данные

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно. Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя натуральными числами b_i, e_i и w_i — номера концов ребра и его вес соответственно ($1 \le b_i, e_i \le n$, $0 \le w_i \le 100\,000$). $n \le 200\,000$, $m \le 200\,000$.

Граф является связным.

Выходные данные

Первая строка выходного файла должна содержать одно натуральное число — вес минимального остовного дерева.

Пример

входные данные	Скопировать
4 4 1 2 1 2 3 2 3 4 5 4 1 4	

выходные данные	Скопировать
7	

K. Алгоритм двух китайцев

ограничение по времени на тест: 6 секунд
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Вам дан взвешенный ориентированный граф, содержащий n вершин и m рёбер. Найдите минимально возможную сумму весов $n - 1$ ребра, которые нужно оставить в графе, чтобы из вершины с номером 1 по этим ребрам можно было добраться до любой другой вершины.

Входные данные

В первой строке даны два целых числа n и m ($1 \le n \le 1\,000$, $0 \le m \le 10\,000$) — количество вершин и ребер в графе.

В следующих m строках даны ребра графа. Ребро описывается тройкой чисел a_i, b_i и w_i ($1 \le a_i, b_i \le n$; $-10^9 \le w_i \le 10^9$) — номер вершины, из которой исходит ребро, номер вершины, в которую входит ребро, и вес ребра.

Выходные данные

Если нельзя оставить подмножество ребер так, чтобы из вершины с номером 1 можно было добраться до любой другой, в единственной строке выведите «NO».

Иначе, в первой строке выведите «YES», а во второй строке выведите минимальную возможную сумму весов ребер, которых необходимо оставить.

Примеры

входные данные	Скопировать
2 1 2 1 10	

выходные данные	Скопировать
NO	

входные данные	Скопировать
4 5 1 2 2 1 3 3 1 4 3 2 3 2 2 4 2	

выходные данные	Скопировать
YES 6	