

Задача А. От списка ребер к матрице смежности, ориентированный граф (1 балл) (!)

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Простой ориентированный граф задан списком ребер, выведите его представление в виде матрицы смежности.

Формат входного файла

Входной файл содержит числа n ($1 \leq n \leq 100$) — число вершин в графе и m ($1 \leq m \leq n(n-1)$) — число ребер. Затем следует m пар чисел — ребра графа.

Формат выходного файла

Выведите в выходной файл матрицу смежности заданного графа.

Пример

input.txt	output.txt
3 4	0 1 0
1 2	0 0 1
2 3	1 1 0
3 1	
3 2	

Задача В. Проверка на неориентированность (1 балл)

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

По заданной квадратной матрице $n \times n$ из нулей и единиц определите, может ли данная матрица быть матрицей смежности простого неориентированного графа.

Формат входного файла

Входной файл содержит число n ($1 \leq n \leq 100$) — размер матрицы, и затем n строк по n чисел, каждое из которых равно 0 или 1 — саму матрицу.

Формат выходного файла

Выведите в выходной файл «YES» если приведенная матрица может быть матрицей смежности простого неориентированного графа и «NO» в противном случае.

Пример

input.txt	output.txt
3 0 1 1 1 0 1 1 1 0	YES
3 0 1 0 1 0 1 1 1 0	NO
3 0 1 0 1 1 1 0 1 0	NO

Задача С. Проверка на наличие параллельных ребер, неориентированный граф (1 балл)

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Неориентированный граф задан списком ребер. Проверьте, содержит ли он параллельные ребра.

Формат входного файла

Входной файл содержит числа n ($1 \leq n \leq 100$) — число вершин в графе и m ($1 \leq m \leq 10\,000$) — число ребер. Затем следует m пар чисел — ребра графа.

Формат выходного файла

Выведите в выходной файл «YES» если граф содержит параллельные ребра и «NO» в противном случае.

Пример

input.txt	output.txt
3 3 1 2 2 3 1 3	NO
3 3 1 2 2 3 2 1	YES

Задача D. Компоненты связности (1 балл)

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

Дан неориентированный граф. Требуется выделить компоненты связности в нем. Подсказка: для решения задачи можно воспользоваться поиском в ширину или поиском в глубину.

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 Е. Кратчайший путь в невзвешенном графе (1 балл)

Имя входного файла: pathbge1.in
Имя выходного файла: pathbge1.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан неориентированный невзвешенный граф. Найдите кратчайшее расстояние от первой вершины до всех вершин.

Формат входного файла

В первой строке входного файла два числа: n и m ($2 \leq n \leq 30000, 1 \leq m \leq 400000$), где n — количество вершин графа, а m — количество ребер.

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной и конечной вершиной. Вершины нумеруются с единицы.

Формат выходного файла

Выведите n чисел — для каждой вершины кратчайшее расстояние до нее.

Пример

pathbge1.in	pathbge1.out
2 1	0 1
2 1	

Задача F. Лабиринт (2 балла)

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Лабиринт представляет собой поле $n \times m$. По некоторым его клеткам ходить можно, а по некоторым — нет. Узник находится в одной из клеток лабиринта и может перемещаться за ход на одну из четырех соседних клеток. Помогите ему дойти до выхода за минимальное число шагов или сообщите, что выйти невозможно.

Формат входного файла

Во входном файле записаны два числа n и m ($0 < n, m < 100$). Далее n строк по m символов описывают лабиринт. Клетка, по которой можно ходить, обозначена символом “.”, клетка, по которой нельзя ходить, обозначена символом “#”. Клетки, обозначенные символами S и T, задают начальную и конечную клетки соответственно.

Формат выходного файла

Если узник может дойти до выхода, выведите в выходной файл минимальное количество действий и далее последовательность команд — символов U, D, R и L, показывающих, в какую сторону нужно идти. Если выйти невозможно, выведите -1.

Пример

input.txt	output.txt
5 4 .S.. ###. T... .#..	7 RRDDLLL

Задача G. Топологическая сортировка (1 балл)

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

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входного файла

В первой строке входного файла даны два натуральных числа N и M ($1 \leq N \leq 100\,000$, $0 \leq M \leq 100\,000$) — 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.

Формат выходного файла

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
3 3 1 2 2 3 3 1	-1

Задача Н. Поиск цикла (2 балла)

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

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

Формат входного файла

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.

Формат выходного файла

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

Пример

cycle.in	cycle.out
2 2 1 2 2 1	YES 2 1
2 2 1 2 1 2	NO