

Задача А. Компоненты связности

Имя входного файла: `matrix.in`
Имя выходного файла: `matrix.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 16 мегабайт

Дан неориентированный невзвешенный граф. Необходимо посчитать количество его компонент связности.

Формат входного файла

В первой строке входного файла содержится одно натуральное число N ($N \leq 100$) — количество вершин в графе. Далее в N строках по N чисел — матрица смежности графа: в i -ой строке на j -ом месте стоит 1, если вершины i и j соединены ребром, и 0, если ребра между ними нет. На главной диагонали матрицы стоят нули. Матрица симметрична относительно главной диагонали.

Формат выходного файла

Вывести одно целое число — искомое количество компонент связности графа.

Пример

matrix.in	matrix.out
6 0 1 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0	3

Задача В. Лесопосадки

Имя входного файла: `tree.in`
Имя выходного файла: `tree.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 16 мегабайт

Дан неориентированный невзвешенный граф. Необходимо определить, является ли он деревом.

Формат входного файла

В первой строке входного файла содержится одно натуральное число N ($N \leq 100$) — количество вершин в графе. Далее в N строках по N чисел — матрица смежности графа: в i -ой строке на j -ом месте стоит 1, если вершины i и j соединены ребром, и 0, если ребра между ними нет. На главной диагонали матрицы стоят нули. Матрица симметрична относительно главной диагонали.

Формат выходного файла

Вывести «YES», если граф является деревом, «NO» иначе.

Пример

tree.in	tree.out
6 0 1 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0	NO
3 0 1 0 1 0 1 0 1 0	YES

Задача С. Поиск цикла

Имя входного файла: `cycle.in`
Имя выходного файла: `cycle.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 16 мегабайт

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 ($1 \leq N \leq 100000, M \leq 100000$) — 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

Задача D. TopSort

Имя входного файла: `topsort.in`
Имя выходного файла: `topsort.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 16 мегабайт

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входного файла

В первой строке входного файла два натуральных числа N и M ($1 \leq N \leq 100000, M \leq 100000$) — 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

topsort.in	topsort.out
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