

Задача А. Обход в глубину

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

Дан неориентированный невзвешенный граф, в котором выделена вершина. Вам необходимо найти количество вершин, лежащих с ней в одной компоненте связности (включая саму выделенную вершину).

Формат входных данных

В первой строке входного файла содержатся два целых числа N и S ($1 \leq S \leq N \leq 100$), где N — количество вершин графа, а S — выделенная вершина. В следующих N строках записано по N чисел — матрица смежности графа, в которой цифра «0» означает отсутствие ребра между вершинами, а цифра «1» — его наличие. Гарантируется, что на главной диагонали матрицы всегда стоят нули.

Формат выходных данных

Выведите одно целое число — искомое количество вершин.

Примеры

<code>dfs.in</code>	<code>dfs.out</code>
5 1 0 1 1 0 0 1 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0	3

Задача В. Компоненты связности

Имя входного файла: `components.in`
Имя выходного файла: `components.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

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

Формат входных данных

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 -й строке на j -м 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 выходных данных

Вывести одно целое число — искомое количество компонент связности графа.

Примеры

<code>components.in</code>	<code>components.out</code>
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 С. Поиск цикла

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 входных данных

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 выходных данных

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

Примеры

<code>cycle.in</code>	<code>cycle.out</code>
2 2 1 2 2 1	YES 1 2
2 2 1 2 1 2	NO

Задача D. Долой списывание!

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

Во время теста Павел Олегович заметил, что некоторые лкшата обмениваются запис-

ками. Сначала он хотел поставить им всем двойки, но в тот день Павел Олегович был добрым, а потому решил разделить лкшат на две группы: списывающих и дающих списывать, и поставить двойки только первым.

У Павла Олеговича записаны все пары лкшат, обменявшихся записками. Требуется определить, сможет ли он разделить лкшат на две группы так, чтобы любой обмен записками осуществлялся от лкшонка одной группы лкшонку другой группы.

Формат входных данных

В первой строке находятся два числа N и M — количество лкшат и количество пар лкшат, обменивающихся записками ($1 \leq N \leq 100$, $0 \leq M \leq \frac{N(N-1)}{2}$). Далее в M строках расположены описания пар лкшат: два различных числа, соответствующие номерам лкшат, обменивающихся записками (нумерация лкшат идёт с 1). Каждая пара лкшат перечислена не более одного раза.

Формат выходных данных

Необходимо вывести ответ на задачу Павла Олеговича. Если возможно разделить лкшат на две группы, выведите «YES»; иначе выведите «NO».

Примеры

bipartite.in	bipartite.out
3 2 1 2 2 3	YES

Примеры

tree.in	tree.out
4 4 1 2 2 3 3 4 4 1	1 2 2 3 3 4

Задача Е. Получи дерево

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

Дан связный неориентированный граф без петель и кратных рёбер, из которого разрешается удалять рёбра. Требуется получить из данного графа дерево.

Формат входных данных

Сначала вводятся два целых числа: N и M ($1 \leq N \leq 100$, $0 \leq M \leq \frac{N(N-1)}{2}$) — количество вершин и рёбер графа соответственно. Далее идёт M пар чисел, задающих рёбра. Гарантируется, что граф связный.

Формат выходных данных

Выведите $N - 1$ пару чисел — рёбра, которые войдут в дерево. Рёбра можно выводить в любом порядке.