Задача А. Светофоры

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

В подземелье M тоннелей и N перекрестков, каждый тоннель соединяет какие-то два перекрестка. Мышиный король решил поставить по светофору в каждом тоннеле перед каждым перекрестком. Напишите программу, которая посчитает, сколько светофоров должно быть установлено на каждом из перекрестков. Перекрестки пронумерованы числами от 1 до N .

Формат входных данных

Во входном файле записано два числа N и M ($0 < N \leq 100$, $0 \leq M \leq \frac{N(N-1)}{2}$). В следующих M строках записаны по два числа i и j ($1 \leq i, j \leq N$), которые означают, что перекрестки i и j соединены тоннелем. Гарантируется, что никакой тоннель не соединяет перекресток сам с собой, и не существует двух различных тоннелей, соединяющих одну и ту же пару перекрестков.

Формат выходных данных

В выходной файл вывести N чисел: k -е число означает количество светофоров на k -м перекрестке.

Примеры

<code>lights.in</code>	<code>lights.out</code>
7 10 5 1 3 2 7 1 5 2 7 4 6 5 6 4 7 5 2 1 5 3	3 3 2 2 5 2 3

Задача В. От матрицы смежности к спискам смежности

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

Простой ориентированный граф задан матрицей смежности. Выведите его представление в виде списков смежности.

Формат входных данных

В первой строке файла находится число N — количество вершин графа ($1 \leq N \leq 100$). Во второй строке и далее — матрица смежности. Гарантируется, что граф не содержит петель.

Формат выходных данных

Выведите N строк — списки смежности графа. В i -й строке сначала выведите количество исходящих из i -й вершины рёбер, а затем — номера вершин, в которые эти рёбра идут, упорядоченные по возрастанию.

Примеры

<code>mtoal.in</code>	<code>mtoal.out</code>
5	1 3
0 0 1 0 0	2 1 3
1 0 1 0 0	1 5
0 0 0 0 1	2 1 2
1 1 0 0 0	2 1 2
1 1 0 0 0	

Задача С. От списков смежности к матрице смежности

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

Формат входных данных

В первой строке входного файла содержится число N — количество вершин ($1 \leq N \leq 100$). Далее идут N строк. В i -й строке содержится описание всех рёбер, исходящих из i -й вершины. Описание начинается количеством исходящих рёбер. Далее следуют номера вершин, в которые эти рёбра идут. Все вершины нумеруются натуральными числами от 1 до N . Гарантируется, что i -й список смежности не содержит числа i , а также все списки не содержат повторяющихся чисел.

Формат выходных данных

Выведите матрицу смежности ориентированного графа.

Примеры

<code>altom.in</code>	<code>altom.out</code>
3	0 1 1
2 2 3	0 0 0
0	0 1 0
1 2	

Задача D. Лесопосадки

Имя входного файла: `tree.in`
Имя выходного файла: `tree.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайта

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

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 -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 выходных данных

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

Примеры

<code>tree.in</code>	<code>tree.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	NO
3 0 1 0 1 0 1 0 1 0	YES

Задача Е. Поиск цикла

Имя входного файла: `cycle.in`
Имя выходного файла: `cycle.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 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 2 1
2 2 1 2 1 2	NO

Задача F. Компоненты связности - 2

Имя входного файла: `matrix2.in`
Имя выходного файла: `matrix2.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайта

Дан неориентированный невзвешенный граф. Необходимо посчитать количество его компонент связности и вывести их.

Формат входных данных

Во входном файле записано два числа N и M ($0 < N \leq 100\,000$), $0 \leq M \leq 100\,000$). В следующих M строках записаны по два числа i и j ($1 \leq i, j \leq N$), которые означают, что вершины i и j соединены ребром.

Формат выходных данных

В первой строчке выходного файла выведите количество компонент связности. Далее выведите сами компоненты связности в следующем формате: в первой строке количество вершин в компоненте, во второй — сами вершины в произвольном порядке.

Примеры

<code>matrix2.in</code>	<code>matrix2.out</code>
6 4	3
3 1	3
1 2	1 2 3
5 4	2
2 3	4 5
	1
	6

Задача G. 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.

Примеры

<code>tree.in</code>	<code>tree.out</code>
4 4	1 2
1 2	2 3
2 3	3 4
3 4	
4 1	

Задача Н. Наводнение

Имя входного файла: `flood.in`
Имя выходного файла: `flood.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

На базе «Берендеевы поляны» все домики соединены между собой дорожками. Каждая дорожка имеет свою высоту над уровнем озера Лель. Каждый домик может быть соединён с другим несколькими дорожками, но дорожек, ведущих из домика в него же, нет.

После сильных дождей озеро выходит из берегов и затапливает дорожки. Дорожка затапливается в том случае, если уровень воды больше или равен её высоте. Поскольку в лагере есть только самокаты и велосипеды, а лодок ещё не закупили, для нормального функционирования лагеря от любого домика до любого другого домика должен существовать путь по незатопленным дорожкам, возможно, по нескольким. Помогите сотрудникам базы определить минимальный уровень воды, при котором найдутся такие два домика, что от одного из них никак нельзя добраться до другого. Гарантируется, что до начала сильных дождей из любого домика можно было попасть в любой другой.

Формат входных данных

В первой строке даны два числа A и B — количество домиков и дорожек соответственно ($2 \leq A \leq 10\,000$, $1 \leq B \leq 20\,000$). Следующие B строк содержат тройки чисел (s_i, f_i, h_i) , где s_i и f_i — номера домиков, которые соединяет i -я дорожка, а h_i — её высота над уровнем озера в миллиметрах ($1 \leq s_i \leq A$, $1 \leq f_i \leq A$, $1 \leq h_i \leq 1\,000\,000$).

Формат выходных данных

Выведите минимальную высоту, на которую должна подняться вода, чтобы в лагере нашлось хотя бы два домика, между которыми нельзя пройти по незатопленным дорожкам.

Примеры

flood.in	flood.out
2 1 1 2 100	100
4 5 1 2 100 1 3 400 2 3 300 2 4 200 3 4 500	300