Задача A. Связность графа

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

Дан граф, содержащий N вершин и M рёбер ($1 \le N \le 1000, 0 \le M \le 7000$). Требуется найти наименьшее число рёбер и эти рёбра, которые нужно добавить, чтобы граф стал связным.

Формат входного файла

Во входном файле записаны сначала числа N и M, затем идёт описание рёбер графа — M пар чисел, где каждая пара описывает начало и конец ребра.

Формат выходного файла

В первую строку вывести единственное число K — минимальное количество рёбер, которое нужно добавить. В следующих K строках выведите по 2 числа — начало и конец нового ребра.

Примеры

connected.in	connected.out
3 1 2 1	1 1 3

Задача B. 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 входного файла

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

Примеры

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

Задача C. Лесопосадки

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

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

Формат входного файла

В первой строке входного файла содержится одно натуральное число N ($N \leqslant 100$) — 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.

Примеры

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 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 записками. 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 входного файла

В первой строке находятся два числа N и M — количество лкшат и количество пар лкшат, обменивающихся записками ($1 \leqslant N \leqslant 100$, $0 \leqslant M \leqslant \frac{N(N-1)}{2}$). 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 выходного файла

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

Задача E. Сделай дерево

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

Дан неориентированный связный граф с кратными ребрами. Найдите максимальный подграф, являющийся деревом.

Формат входного файла

В первой строке даны количество вершин N и ребер M ($1 \leq N, M \leq 100000$). В следующих M строках даны пары вершин $v1, v2$, являющимися концами ребер($1 \leq v1, v2 \leq N$).

Формат выходного файла

В первой строке выведите количество ребер в дереве. В каждых последующих строках выведите список ребер в формате, аналогичному входному файлу.

Примеры

maketree.in	maketree.out
3 3 1 2 2 3 1 3	2 2 3 1 2
2 2 1 2 1 2	1 1 2

Задача F. Построение

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

Группа солдат-новобранцев прибыла в армейскую часть N666. После знакомства с прапорщиком стало очевидно, что от работ на кухне по очистке картофеля спасти солдат может только чудо.

Прапорщик, будучи не в состоянии запомнить фамилии, пронумеровал новобранцев от 1 до N. После этого он велел им построиться по росту (начиная с самого высокого). С этой несложной задачей могут справиться даже совсем необученные новобранцы, да вот беда, прапорщик уверил себя, что знает про некоторых солдат, кто из них кого выше, и это далеко не всегда соответствует истине.

После трех дней обучения новобранцам удалось выяснить, что знает (а точнее, думает, что знает) прапорщик. Помогите им, используя эти знания, построиться так, чтобы товарищ прапорщик остался доволен.

Формат входного файла

Сначала на вход программы поступают числа N и M ($1 < N \leqslant 100$, $1 \leqslant M \leqslant 5000$) — количество солдат в роте и количество пар солдат, про которых прапорщик знает, кто из них выше. Далее идут эти пары чисел A и B по одной на строке ($1 \leqslant A, B \leqslant N$), что означает, что, по мнению прапорщика, солдат A выше, чем B.

Формат выходного файла

В первой строке выведите «`Yes`» (если можно построиться так, чтобы прапорщик остался доволен) или «`No`» (если нет). После ответа «`Yes`» на следующей строке выведите N чисел, разделенных пробелами, — одно из возможных построений.

Примеры

formation.in	formation.out
5 4 1 3 1 4 4 3 5 2	Yes 5 2 1 4 3