

Обход в глубину (Depth-First-Search) и родственные задачи.

А. Количество вершин в компоненте связности

Дан неориентированный граф. Требуется найти количество вершин, лежащих в одной компоненте связности с данной вершиной (считая эту вершину).

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

Выведите одно целое число — искомое количество вершин.

Input	Output
3 1 0 1 1 1 0 0 1 0 0	3

В. Проверить является ли граф деревом

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

В первой строке входного файла содержится одно натуральное число N ($N \leq 100$) — количество вершин в графе. Далее в N строках записано по N чисел — матрица смежности графа.

На главной диагонали матрицы стоят нули. Матрица симметрична относительно главной диагонали.

Вывести YES, если граф является деревом, и NO в противном случае.

Input	Output
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

С. Компоненты связности

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

Во входном файле записано два числа N и M ($0 < N \leq 100000$, $0 \leq M \leq 100000$). 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.

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.

Input	Output
6 4 3 1 1 2 5 4 2 3	3 3 1 2 3 2 4 5 1 6

Д. Цикл в ориентированном графе

Дан ориентированный граф. Требуется определить, есть ли в нём цикл.

Циклом называется простой путь $v_1 \rightarrow v_2, v_2 \rightarrow v_3, \dots, v_{n-1} \rightarrow v_n$ ($v_i \neq v_j$), такой, что существует также ребро $v_n \rightarrow v_1$.

В первой строке входного файла содержится одно натуральное число N ($N \leq 50$) — количество вершин в графе. Далее в N строках по N чисел — матрица смежности графа. Гарантируется, что на диагонали матрицы будут стоять нули.

Выведите NO, если в заданном графе цикла нет, и YES, если он есть.

Input	Output
3 0 1 0 0 0 1 0 0 0	NO
3 0 1 0 0 0 1 1 0 0	YES

Е. Цикл в неориентированном графе

Дан неориентированный граф. Требуется определить, есть ли в нём цикл, и, если есть, вывести его.

Циклом в неориентированном графе называется простой путь, содержащий попарно различные рёбра $v_1 \leftrightarrow v_2, v_2 \leftrightarrow v_3, \dots, v_{n-1} \leftrightarrow v_n$ ($v_i \neq v_j$), такой, что существует также ребро $v_n \leftrightarrow v_1$.

В первой строке дано одно число N ($1 \leq N \leq 500$) — количество вершин в графе. Далее в N строках задан сам граф матрицей смежности.

Если в исходном графе нет цикла, то выведите NO. Иначе, в первой строке выведите YES, во второй строке выведите число K — количество вершин в цикле, а в третьей строке выведите K различных чисел — номера вершин, которые принадлежат циклу в порядке его обхода (обход можно начинать с любой вершины цикла). Если циклов несколько — выведите любой.

Input	Output
3 0 1 1 1 0 1 1 1 0	YES 3 3 2 1
4 0 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0	NO

Ф. Проверка графа на двудольность

На банкет были приглашены N Очень Важных Персон (ОВП). Были поставлены 2 стола. Столы достаточно большие, чтобы все посетители банкета могли сесть за любой из них. Проблема заключается в том, что некоторые ОВП не ладят друг с другом и не могут сидеть за одним столом. Вас попросили определить, возможно ли всех ОВП рассадить за двумя столами.

В первой строке входных данных содержатся два числа: N и M ($1 \leq N, M \leq 100$), где N — количество ОВП, а M — количество пар ОВП, которые не могут сидеть за одним столом. В следующих M строках записано по 2 числа — пары ОВП, которые не могут сидеть за одним столом.

Если способ рассадить ОВП существует, то выведите YES в первой строке и номера ОВП, которых необходимо посадить за первый стол, во второй строке. В противном случае в первой и единственной строке выведите NO.

Input	Output
3 2 1 2 1 3	YES 1