

Задача А. Обход в ширину

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

Дан неориентированный граф. В нём необходимо найти расстояние от одной заданной вершины до другой.

Формат входных данных

В первой строке входного файла содержатся три натуральных числа N , S и F ($1 \leq S, F \leq N \leq 100$) — количество вершин в графе и номера начальной и конечной вершин соответственно. Далее в N строках задана матрица смежности графа. Если значение в j -м элементе i -й строки равно 1, то в графе есть направленное ребро из вершины i в вершину j .

Формат выходных данных

В единственной строке должно находиться минимальное расстояние от начальной вершины до конечной. Если пути не существует, выведите 0.

Примеры

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

Задача В. Топологическая сортировка

Имя входного файла: `topsort.in`
Имя выходного файла: `topsort.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входных данных

В первой строке входного файла даны два целых числа N и M ($1 \leq N \leq 100\,000, 0 \leq M \leq 100\,000$) — количества 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

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести «-1».

Примеры

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

Задача С. Поиск цикла

Имя входного файла: `cycle2.in`
Имя выходного файла: `cycle2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан ориентированный невзвешенный граф без кратных рёбер. 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 ($1 \leq N \leq 100\,000, M \leq 100\,000$) — количества вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин.

Формат выходных данных

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

Примеры

<code>cycle2.in</code>	<code>cycle2.out</code>
2 2 1 2 2 1	YES 1 2
2 1 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 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 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	YES
1 2	
2 3	

Задача Е. Кратчайший путь

Имя входного файла: dag-shortpath.in
Имя выходного файла: dag-shortpath.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан ориентированный взвешенный ациклический граф. Требуется найти в нем кратчайший путь из вершины s в вершину t .

Формат входных данных

Первая строка входного файла содержит четыре целых числа n , m , s и t — количество вершин, дуг графа, начальная и конечная вершина соответственно. Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается тремя натуральными числами b_i , e_i и w_i — началом, концом и длиной дуги соответственно ($1 \leq b_i, e_i \leq n$, $|w_i| \leq 1000$).

Входной граф не содержит циклов и петель.

$1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$.

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — длину кратчайшего пути из s в t . Если пути из s в t не существует, выведите «Unreachable».

Примеры

dag-shortpath.in	dag-shortpath.out
2 1 1 2 1 2 -10	-10
2 1 2 1 1 2 -10	Unreachable

Задача F. 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 входных данных

В первой строке даны количество вершин N и ребер M ($1 \leq N, M \leq 100\,000$). В следующих M строках даны пары вершин v_1, v_2 , являющимися концами ребер ($1 \leq v_1, v_2 \leq N$).

Формат выходных данных

В первой строке выведите количество ребер в дереве. В каждом последующем строке 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