

Задача А. 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 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

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 В. 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

cycle2.in	cycle2.out
2 2 1 2 2 1	YES 1 2
2 1 1 2	NO

Задача С. 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 1 2 2 3	YES

Задача D. TopSort

Имя входного файла: `topsort.in`
Имя выходного файла: `topsort.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входных данных

В первой строке входного файла даны два натуральных числа N и M ($1 \leq N \leq 10^5, 1 \leq M \leq 10^5$) — 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

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, требуется 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
3 3 1 2 2 3 3 1	-1

Задача Е. Удаление клеток

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

Из прямоугольного листа клетчатой бумаги (M строк, N столбцов) удалили некоторые клетки. На сколько кусков распадётся оставшаяся часть листа? Две клетки не распадаются, если они имеют общую сторону.

Формат входных данных

В первой строке находятся числа M и N , в следующих M строках - по N символов. Если клетка не была вырезана, этому соответствует знак `#`, если вырезана - точка. $1 \leq M, N \leq 100$.

Формат выходных данных

Вывести одно число.

Примеры

<code>cell-delete.in</code>	<code>cell-delete.out</code>
<pre>5 10 ##.#####. .#.#.#.... ###.##.#. ..##.....# .###.#####</pre>	<pre>5</pre>

Задача F. Борода админа

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Известный спелеолог, автор многотомного фолианта по спелеологии, Дональд К. исследовал пещеру «Борода админа». Пещера называется так, потому что похожа на бороду админа следующим свойством: от любого грота до любого другого существует ровно один простой путь по лазам. (С этого момента будем называть гроты — вершинами, лазы — ребрами, а пещеру — деревом, причем подвешенным за первую вершину. Надеемся, Дональд К. сможет разобраться в нашей терминологии.)

Дональд К. использовал следующую процедуру обхода пещеры, начиная с грота, в котором расположен вход, то есть с первой вершины:

- построить предположение о том, сколько вершин в поддереве этой вершины, записать в блокнот,
- для каждой из ещё не посещённых вершин, соседних с данной,
- пойти в нее, применить данную процедуру, вернуться,
- убедиться, что предположение было верным: Дональд К. не ошибается никогда.

Вы нашли блокнот с записями Дональда К. Восстановите план пещеры «Борода админа».

Формат входных данных

В первой строке содержится количество прогнозов Дональда К. $N \leq 100\,000$. Во второй строке содержатся прогнозы Дональда К. — натуральные числа, не превосходящие 100 000.

Формат выходных данных

Выведите дерево, в котором описанная процедура дает такие результаты. А именно, необходимо вывести $N - 1$ число: для каждой вершины v , кроме стартовой, необходимо вывести номер той вершины, из которой Дональд К. попал в вершину v впервые.

Если таких деревьев несколько, выведите любое.

Гарантируется, что хотя бы одно такое дерево существует.

Примеры

<code>stdin</code>	<code>stdout</code>
5	1 1 3 3
5 1 3 1 1	

Задача G. Кратчайший путь

Имя входного файла: `dag-shortpath.in`
Имя выходного файла: `dag-shortpath.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

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.

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

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

<code>dag-shortpath.in</code>	<code>dag-shortpath.out</code>
2 1 1 2 1 2 -10	-10
2 1 2 1 1 2 -10	Unreachable