Содержание

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

В некоторых задачах большой ввод и вывод. Пользуйтесь быстрым вводом-выводом.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) переопределение стандартного аллокатора ускорит вашу программу.

Обратите внимание на `GNU C++` компиляторы с суффиксом `inc`, они позволяют пользоваться дополнительной библиотекой. Под ними можно сдать вот это.

Must have

Задача 4A. Максимальный поток [0.3 sec, 256 mb]

Вам задан ориентированный граф G. Каждое ребро имеет некоторую пропускную способность. Найдите максимальный поток между вершинами 1 и n.

Формат входных данных

Первая строка входного файла содержит n и m — число вершин и ребер в графе ($2 \leqslant n \leqslant 500$, $1 \leqslant m \leqslant 10\,000$). Последующие строки описывают ребра. Каждое ребро задается тремя числами: начальная вершина ребра, конечная вершина ребра и пропускная способность ребра. Пропускные способности не превосходят 10^9.

Формат выходных данных

Выведите величину максимального потока между вершинами 1 и n.

Далее для каждого ребра выведите величину потока, текущую по этому ребру.

Примеры

stdin	stdout
4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1	3.0 1.0 2.0 1.0 2.0 1.0

Замечание

Диниц с масштабированием точно получает `OK` с огромным запасом.

Обязательные задачи

Задача 4B. Perspective [0.2 sec, 256 mb]

Баскетбол. NBA. Несколько команд играют турнир. Команды разбиты на дивизионы. Есть игры внутри дивизиона и между командами из разных дивизионов. Ничей не бывают, в каждой игре кто-то выигрывает, кто-то проигрывает. У вас есть любимая команда. Вы знаете всё про дивизион, в котором эта команда играет. Какие-то игры уже сыграны. Про каждую команду дивизиона вам известно, сколько побед уже одержала эта команда и сколько игр ещё предстоит ей сыграть (включая и игры внутри дивизиона, и игры вне дивизиона). Про каждую пару команд дивизиона известно, сколько ещё игр им предстоит сыграть друг с другом. Определите, могут ли все оставшиеся игры быть сыграны так, чтобы в результате у вашей любимой команды было побед не меньше чем у любой другой в её дивизионе?

Формат входных данных

На первой строке число команд в дивизионе n ($2 \leqslant n \leqslant 20$). Ваша любимая команда имеет номер 1. Следующая строка содержит n чисел, i-е число обозначает количество побед у i-й команды. Следующая строка содержит n чисел, i-е число обозначает количество предстоящих игр у i-й команды. Далее n строк содержат по n чисел, j-е число на i-й строке обозначает, количество предстоящих матчей между i-й и j-й командой дивизиона. Гарантируется, что матрица симметрична, и на диагонали стоят нули. Все числа целые, неотрицательные, до 10 000.

Формат выходных данных

Выведите `YES` или `NO`.

Пример

stdin	stdout
3 1 2 2 1 1 1 0 0 0 0 0 0 0 0 0	YES
3 1 2 2 1 1 1 0 0 0 0 0 1 0 1 0	NO

Замечание

Что-то похожее разобрано на старой практике.

Задача 4C. Охлаждение реактора [0.2 sec, 256 mb]

Известная террористическая группа под руководством знаменитого террориста Бен Гадена решила построить атомный реактор для получения оружейного плутония. Вам, как компьютерному гению этой группы, поручили разработать систему охлаждения реактора.

Система охлаждения реактора представляет собой набор труб, соединяющих узлы. По трубам течет жидкость, причем для каждой трубы строго определено направление, в котором она должна по ней течь. Узлы системы охлаждения занумерованы от 1 до N. Система охлаждения должна быть спроектирована таким образом, чтобы для каждого узла за единицу времени количество жидкости, втекающей в узел, было равно количеству жидкости, вытекающей из узла. То есть если из i-го узла в j-ый течет f_{ij} единиц жидкости за единицу времени (если из i в j нет трубы, то положим $f_{ij} = 0$), то для каждого узла i должно выполняться

$$\sum_{j=1}^{N} f_{ij} = \sum_{j=1}^{N} f_{ji}$$

У каждой трубы имеется пропускная способность c_{ij}. Кроме того, для обеспечения достаточного охлаждения требуется, чтобы по трубе протекало не менее l_{ij} единиц жидкости за единицу времени. То есть для трубы, ведущей из i-го узла в j-ый должно выполняться $l_{ij} \leqslant f_{ij} \leqslant c_{ij}$.

Вам дано описание системы охлаждения, выясните, каким образом можно пустить жидкость по трубам, чтобы выполнялись все указанные условия.

Формат входных данных

Первая строка входного файла содержит числа N и M — количество узлов и труб ($1 \leqslant N \leqslant 200$). Следующие M строк содержат описание труб. Каждая строка содержит четыре целых числа i, j, l_{ij} и c_{ij}. Любые два узла соединены не более чем одной трубой, если есть труба из i в j, то нет трубы из j в i, никакой узел не соединен трубой сам с собой, $0 \leqslant l_{ij} \leqslant c_{ij} \leqslant 10^5$.

Формат выходных данных

Если решение существует, выведите на первой строке выходного файла слово `YES`. Затем выведите M чисел — количество жидкости, которое должно течь по трубам, числа должны быть выведены в том порядке, в котором трубы заданы во входном файле. Если решения не существует, выведите `NO`.

Пример

stdin	stdout
4 6 1 2 1 2 2 3 1 2 3 4 1 2 4 1 1 2 1 3 1 2 4 2 1 2	NO
4 6 1 2 1 3 2 3 1 3 3 4 1 3 4 1 1 3 1 3 1 3 4 2 1 3	YES 1 2 3 2 1 1

Замечание

Разобрано на новой практике.

Дополнительные задачи

Задача 4D. Sociology [0.5 sec, 256 mb]

Vasya works for the RISK (Research Institute of Sociological tasKs). He is studying relationships between software engineers working freelance and their managers. Vasya considers several jobs assigned to programmers. Each job is to be done by one engineer and is to be managed by one manager.

Vasya calls a subset A of managers *excessive* if the subset of engineers having common jobs with at least one of managers from A has lesser cardinality than $|A|$. His hypothesis is that a system having no excessive subsets of managers is more stable and produces less pressure to the workers.

Your task is to find an excessive subset or say that it is impossible.

Формат входных данных

Input consists of no more than 10 test cases. The first line of each test case contains two integers N_e and N_m — the number of engineers and managers, respectively ($1 \leqslant N_e, N_m \leqslant 10^4$). The next line contains a single integer N_j — the number of jobs ($1 \leqslant N_j \leqslant 10^5$). Then N_j lines follow, each containing two integers e_i and m_i — numbers of engineer and manager assigned to job number i.

Формат выходных данных

For each test case, output either an excessive subset of managers or a message that it does not exist. Adhere to the sample output below as close as possible.

Пример

stdin
3 3 6 1 2 1 3 2 3 2 1 3 1 3 2 2 3 2 1 3 2 2 1 3 3 1 1 1 2 1 3
stdout
Case #1: There is no excessive subset of managers. Case #2: Manager 1 forms an excessive subset. Case #3: Managers 1, 2 and 3 form an excessive subset.

Замечание

Можно или долго и муторно упихивать ногами Куна, или попробовать Хопкрофта-Карпа. Для скорости работы в Хопкрофте-Карпе важно `dfs` писать как в Куне, а не как в Форде-Фалкерсоне.

Задача 4E. Живопись [1 sec, 256 mb]

В стране Олимпия очень развита живопись. Картиной считается любой прямоугольник, который состоит из черных и белых единичных квадратов. Художник Олимпус решил радикально улучшить свои картины. Для этого он планирует к белому и черному цветам добавить еще и серый оттенок. По его задумке, граница между каждыми черным и белым квадратом должна содержать серую линию, чтобы образовался эффект плавного перехода.

Однако, перед началом работы, он обнаружил, что серая краска очень дорого стоит. Чтобы сэкономить деньги художник решил оценить, не выгоднее ли сначала перекрасить некоторые белые квадраты в черные, а черные в белые для того, чтобы минимизировать расходы на краску.

Напишите программу, которая по информации о существующей картине определяет минимальную сумму денег, которые понадобятся на ее улучшение.

Формат входных данных

Первая строка входного файла содержит пять натуральных чисел N, M, w, b, g. $1 \leqslant N, M \leqslant 70$ — высота и ширина картины, $1 \leqslant w, b, g \leqslant 1000$ — цена рисования одного белого единичного квадрата, черного единичного квадрата и серой линии единичной длины, соответственно. Далее следует N строк, каждая из которых состоит из M литер. Литера B соответствует черному квадрату, а W — белому.

Формат выходных данных

Единственная строка выходного файла должна содержать одно целое число, которое есть минимальной суммой затрат на улучшение картины.

Пример

stdin	stdout
3 2 3 3 2 BB WW WB	7

Задача 4F. Равномерный поток [1 sec, 256 mb]

Дана система из узлов и труб, по которым может течь вода. Для каждой трубы известна наибольшая скорость, с которой вода может протекать через нее. Известно, что вода течет по трубам таким образом, что за единицу времени в каждый узел (за исключением двух — источника и стока) втекает ровно столько воды, сколько из него вытекает. Более того, известно, что для любой пары узлов (включая источник и сток) сумма скоростей течения воды вдоль любого пути, их соединяющего, постоянна для данной пары узлов. Сумма берется таким образом, что если труба представлена в пути против направления движения воды в ней, то соответствующее слагаемое берется со знаком минус.

Ваша задача — найти наибольшее количество воды, которое за единицу времени может протекать между источником и стоком, а также скорость течения воды по каждой из труб.

Трубы являются двусторонними, то есть вода в них может течь в любом направлении. Между любой парой узлов может быть более одной трубы.

Формат входных данных

В первой строке записано натуральное число N — количество узлов в системе ($2 \leqslant N \leqslant 100$). Известно, что источник имеет номер 1, а сток номер N. Во второй строке записано натуральное M ($1 \leqslant M \leqslant 5000$) — количество труб в системе. Далее в M строках идет описание труб. Каждая труба задается тройкой целых чисел A_i, B_i, C_i, где A_i, B_i — номера узлов, которые соединяет данная труба ($A_i \neq B_i$), а C_i ($0 \leqslant C_i \leqslant 10^4$) — наибольшая допустимая скорость течения воды через данную трубу.

Формат выходных данных

В первой строке выведите наибольшее количество воды, которое протекает между источником и стоком за единицу времени. Далее выведите M строк, в каждой из которых выведите скорость течения воды по соответствующей трубе. Если направление не совпадает с порядком узлов, заданным во входных данных, то выводите скорость со знаком минус. Числа выводите с точностью 10^{-3}.

Пример

stdin	stdout
2 1 1 2 1	1.0000000000 1.0000000000
3 2 1 2 1 2 3 2	1.0000000000 1.0000000000 1.0000000000