

Содержание

Must have	2
Задача 5А. Максимальный поток [0.3 sec, 256 mb]	2
Обязательные задачи	3
Задача 5В. Паросочетание в случайном графе [0.1 sec, 256 mb]	3
Задача 5С. Perspective [0.2 sec, 256 mb]	4
Задача 5D. Толстые хоббиты [0.2 sec, 256 mb]	5
Задача 5Е. Охлаждение реактора [0.2 sec, 256 mb]	6
Дополнительные задачи	8
Задача 5F. Игры на графе [0.5 sec, 256 mb]	8
Задача 5G. Мошенники [0.5 sec, 256 mb]	9
Задача 5H. Sociology [0.5 sec, 256 mb]	10
Задача 5I. Живопись [1 sec, 256 mb]	11

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 5А. 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 ($2 \leq n \leq 500$, $1 \leq m \leq 10\,000$). 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	3.0
1 2 1	1.0
1 3 2	2.0
3 2 1	1.0
2 4 2	2.0
3 4 1	1.0

Замечание

Диниц с масштабированием точно получает ОК с огромным запасом.

Обязательные задачи

Задача 5В. Паросочетание в случайном графе [0.1 sec, 256 mb]

Дан неориентированный граф, нужно выделить в нем максимальное паросочетание.

Гарантируется, что граф сгенерирован следующим методом: фиксируется количество вершин N и количество ребер M , далее говорится, что все концы всех ребер — случайные числа от 1 до N

Соответственно, граф может содержать петли и кратные ребра.

Формат входных данных

N ($1 \leq N \leq 200$) и M ($0 \leq M \leq 1000$).

Формат выходных данных

Выведите на первой строке количество ребер в паросочетание. Далее собственно ребра, каждое на отдельной строке, в произвольном порядке.

Пример

stdin	stdout
5 7	2
1 2	5 4
2 3	1 3
3 1	
3 4	
4 5	
1 2	
5 3	

Замечание

На случайном графе отлично работают простые «Куноподобные» способы.

Задача 5С. 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 ($2 \leq n \leq 20$). 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 5D. Толстые хоббиты [0.2 sec, 256 mb]

Ни один хоббит не в состоянии в одиночку противостоять полчищам Мордора... В последний поход против Мордора Гэндальф решил отправить N хоббитов из Шира. Но часть хоббитов наотрез отказалась, жалуюсь на то, что другие хоббиты наверняка будут дразнить их толстыми. После опроса всех хоббитов оказалось, что любой хоббит отказывается принять участие в походе в том случае, если с ним в поход выступит хотя бы один хоббит с меньшим весом. К счастью для Средиземья, не все хоббиты знают свой точный вес. В Шире были всего одни весы чашечного типа, позволяющие для пары хоббитов определить, какой хоббит тяжелее. Некоторые пары хоббитов взвешивались на этих весах. Всем хоббитам известен результат всех взвешиваний. Гэндальф абсолютно уверен, что в Шире нет двух хоббитов одного веса. Он заинтересован в том, чтобы отряд состоял из наибольшего количества хоббитов. Однако найти наибольшее множество хоббитов, среди которых ни один не считает себя тяжелее другого, оказалось не так-то просто. Подскажите Гэндальфу, на сколько хоббитов он может рассчитывать. Помните при этом, что хоббиты умные существа и знают, что если Сэм тяжелее Пиппина, а Пиппин тяжелее Фродо, то Сэм и подавно будет тяжелее Фродо.

Формат входных данных

В первой строке дано целое число N — количество хоббитов ($2 \leq N \leq 100$). Все хоббиты пронумерованы целыми числами от 1 до N . В следующих N строках записана матрица размера $N \times N$. Если i -й и j -й хоббит взвешивались на чашечных весах и оказалось, что i -й хоббит тяжелее, то в i -й строке матрицы на j -й позиции стоит единица. Во всех остальных случаях в матрице стоят нули.

Формат выходных данных

В первой строке выведите размер наибольшего множества хоббитов, готового выступить в поход, во второй строке перечислите номера хоббитов из этого множества через пробел.

Пример

stdin	stdout
2 0 1 0 0	1 2
3 0 0 0 0 0 0 0 0 0	3 1 2 3

Замечание

Антицепь.

Задача 5Е. 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 $l_{ij} \leq f_{ij} \leq c_{ij}$.

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 ($1 \leq N \leq 200$). 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, $0 \leq l_{ij} \leq c_{ij} \leq 10^5$.

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 5F. Игры на графе [0.5 sec, 256 mb]

Коля и Петя любят играть в следующую игру на лекциях по теории сложности. Они рисуют двудольный граф G на листке бумаги и ставят фишку в одну из вершин графа. Далее они ходят по очереди, Коля ходит первым.

Ход состоит из перемещения фишки по ребру в графе. После перемещения фишки вершина, из которой фишка только что ушла, удаляется из графа вместе со всеми инцидентными ей рёбрами. Проигрывает игрок, который не может ходить.

Вам дан граф, который нарисовали Коля и Петя. Для каждой вершины графа определите, кто выиграет, если изначально фишка находится в этой вершине. Предполагайте, что и Коля, и Петя играют оптимально.

Формат входных данных

Первая строка файла содержит три целых числа n_1, n_2, m – число вершин в одной доли, второй доли и число рёбер, соответственно ($1 \leq n_1, n_2 \leq 500, 0 \leq m \leq 50\,000$). Следующие m строк описывают рёбра графа – каждая строка содержит номера вершин, которые рёбро соединяет. Вершины в каждой из долей нумеруются с 1.

Формат выходных данных

Выведите две строки. На первой строке выведите n_1 символов, i -й символ должен быть 'N', если при начале игры из i -й вершины первой доли выиграет Коля, или 'P', если выиграет Петя. Аналогично во второй строке выведите n_2 символов – результаты игры для второй доли графа.

Пример

stdin	stdout
3 3 5	NPP
1 1	NPP
1 2	
1 3	
2 1	
3 1	

Замечание

Да, да, это тоже про паросочетания.

Задача 5G. Мошенники [0.5 sec, 256 mb]

Студенты придумали N тем для курсовых, название каждой состоит из двух слов из заглавных букв английского алфавита. Среди студентов есть читеры, которые вместо того, чтобы придумывать собственные названия, взяли оба слова из уже придуманных их коллегами тем. При этом в качестве первого слова они взяли первое слово одной из тем, а в качестве второго второе. Вам даны все N названий в произвольном порядке, определите максимальное число читеров среди студентов.

Формат входных данных

Первая строка содержит число тестов T . Каждый тест содержит N и N пар слов – названия тем. Все названия различны. $1 \leq T \leq 100$, $1 \leq N \leq 1000$, длины слов от 1 до 20.

Формат выходных данных

Выведите максимальное число читеров. Формат вывода смотрите в примере.

Примеры

stdin	stdout
3	Case #1: 1
3	Case #2: 0
HYDROCARBON COMBUSTION	Case #3: 0
QUAIL BEHAVIOR	
QUAIL COMBUSTION	
3	
CODE JAM	
SPACE JAM	
PEARL JAM	
2	
INTERGALACTIC PLANETARY	
PLANETARY INTERGALACTIC	

Задача 5H. 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 ($1 \leq N_e, N_m \leq 10^4$). The next line contains a single integer N_j — the number of jobs ($1 \leq N_j \leq 10^5$). 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 упикивать 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 51. 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 каждым 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 . $1 \leq N, M \leq 70$ — высота и ширина картины, $1 \leq w, b, g \leq 1000$ — 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