

Задача А. Максимальный поток

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

Задан ориентированный граф, каждое ребро которого обладает целочисленной пропускной способностью. Найдите максимальный поток из вершины с номером 1 в вершину с номером n .

Формат входных данных

Первая строка входного файла содержит n и m — количество вершин и количество ребер графа ($2 \leq n \leq 100$, $1 \leq m \leq 1000$). Следующие m строк содержат по три числа: номера вершин, которые соединяет соответствующее ребро графа и его пропускную способность. Пропускные способности не превосходят 10^5 .

Формат выходных данных

В выходной файл выведите одно число — величину максимального потока из вершины с номером 1 в вершину с номером n .

Примеры

maxflow.in	maxflow.out
4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1	3

Задача В. Максимальный поток

Имя входного файла: `flow2.in`
Имя выходного файла: `flow2.out`
Ограничение по времени: 0.5 секунда
Ограничение по памяти: 256 мегабайт

Задан ориентированный граф, каждое ребро которого обладает целочисленной пропускной способностью. Найдите максимальный поток из вершины с номером 1 в вершину с номером 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 .

Примеры

flow2.in	flow2.out
4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1	3

Задача С. Разрез

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан неориентированный граф. Найдите минимальный разрез между вершинами 1 и n .

Формат входных данных

На первой строке входного файла содержится n ($2 \leq n \leq 100$) — число вершин в графе и m ($0 \leq m \leq 400$) — количество ребер. На следующих m строках входного файла содержится описание ребер. Ребро описывается номерами вершин, которые оно соединяет, и его пропускной способностью (положительное целое число, не превосходящее 10 000 000), при этом никакие две вершины не соединяются более чем одним ребром.

Формат выходных данных

На первой строке выходного файла должны содержаться количество ребер в минимальном разрезе и их суммарная пропускная способность. На следующей строке выведите возрастающую последовательность номеров ребер (ребра нумеруются в том порядке, в каком они были заданы во входном файле).

Примеры

стандартный ввод	стандартный вывод
6 8 1 2 3 1 3 3 2 4 2 2 5 2 3 4 2 3 5 2 5 6 3 4 6 3	2 6 1 2

Задача D. Чокнутый профессор

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

В Университете города М. проводят эксперимент. Преподаватели сами решают, что они будут читать в рамках того или иного курса. И вот преподаватель математического анализа (в простонародье — матана) оценил по некоторым критериям все известные ему темы в данном курсе. В результате этой ревизии каждой теме сопоставлено некоторое целое число (возможно, отрицательное) — полезность данной темы. Профессор хочет максимизировать суммарную полезность прочитанных им тем, но не все так просто. Для того что бы студенты поняли некоторые темы, необходимо, чтобы были прочитаны так же некоторые другие темы, так как некоторые доказательства базируются на фактах из других тем. Однако если существует цикл из зависимостей тем, то их все можно прочитать, и на качестве понимания материала студентами это не скажется.

Вас попросили составить список тем, которые профессор должен прочитать, таким образом, чтобы студенты все поняли, и суммарная полезность курса была максимальна.

Формат входных данных

Первая строка входного файла содержит одно число — N ($1 \leq N \leq 200$). Вторая строка содержит N целых чисел, не превосходящих по модулю 1000 — полезности каждой темы. Далее следуют N строк с описанием зависимостей тем. Каждое описание начинается количеством тем, которые необходимо понять для понимания данной темы. Потом следуют номера этих тем, разделенные пробелами.

Формат выходных данных

Выведите единственное число — максимально возможную суммарную полезность прочитанного материала.

Примеры

stdin	stdout
4 -1 1 -2 2 0 1 1 2 4 2 1 1	2
3 2 -1 -2 2 2 3 0 0	0

Задача Е. Улиточки

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2.5 секунды
Ограничение по памяти: 256 мегабайт

Две улиточки Маша и Петя сейчас находятся на лужайке с абрикосами и хотят добраться до своего домика. Лужайки пронумерованы числами от 1 до n и соединены дорожками (может быть несколько дорожек соединяющих две лужайки, могут быть дорожки, соединяющие лужайку с собой же). Ввиду соображений гигиены, если по дорожке проползла улиточка, то вторая по той же дорожке уже ползти не может. Помогите Пете и Маше добраться до домика.

Формат входных данных

В первой строке файла записаны четыре целых числа — n , m , a и h (количество лужаек, количество дорог, номер лужайки с абрикосами и номер домика).

В следующих m строках записаны пары чисел. Пара чисел (x, y) означает, что есть дорожка с лужайки x до лужайки y (из-за особенностей улиток и местности дорожки односторонние).

Ограничения: $2 \leq n \leq 10^5$, $0 \leq m \leq 10^5$, $a \neq h$

Формат выходных данных

Если существует решение, то выведите YES и на двух отдельных строчках сначала путь для Машеньки (т.к. дам нужно пропускать вперед), затем путь для Пети. Если решения не существует, выведите NO. Если решений несколько, выведите любое.

Примеры

stdin	stdout
3 3 1 3	YES
1 2	1 3
1 3	1 2 3
2 3	

Задача F. Великая стена

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

У короля Людовика двое сыновей. Они ненавидят друг друга, и король боится, что после его смерти страна будет уничтожена страшными войнами. Поэтому Людовик решил разделить свою страну на две части, в каждой из которых будет властвовать один из его сыновей. Он посадил их на трон в города A и B , и хочет построить минимально возможное количество фрагментов стены таким образом, чтобы не существовало пути из города A в город B .

Страну, в которой властвует Людовик, можно упрощенно представить в виде прямоугольника $m \times n$. В некоторых клетках этого прямоугольника расположены горы, по остальным же можно свободно перемещаться. Кроме этого, ландшафт в некоторых клетках удобен для строительства стены, в остальных же строительство невозможно.

При поездках по стране можно перемещаться из клетки в соседнюю по стороне, только если ни одна из этих клеток не содержит горы или построенного фрагмента стены.

Формат входных данных

В первой строке входного файла содержатся числа m и n ($1 \leq m, n \leq 50$). Во второй строке заданы числа k и l , где $0 \leq k, l, k + l \leq mn - 2$, k — количество клеток, на которых расположены горы, а l — количество клеток, на которых можно строить стену. Естественно, что на горах строить стену нельзя. Следующие k строк содержат координаты клеток с горами x_i и y_i , а за ними следуют l строк, содержащие координаты клеток, на которых можно построить стену — x_j и y_j . Последние две строки содержат координаты городов A (x_A и y_A) и B (x_B и y_B) соответственно. Среди клеток, описанных в этих $k + l + 2$ строках, нет двух совпадающих. Гарантируется, что $1 \leq x_i, x_j, x_A, x_B \leq m$ и $1 \leq y_i, y_j, y_A, y_B \leq n$.

Формат выходных данных

В первой строке выходного файла должно быть выведено минимальное количество фрагментов стены F , которые необходимо построить. В последующих F строках необходимо вывести один из возможных вариантов застройки.

Если невозможно произвести требуемую застройку, то необходимо вывести в выходной файл единственное число -1 .

Примеры

стандартный ввод	стандартный вывод
5 5	3
3 8	1 3
3 2	2 3
2 4	3 1
3 4	
3 1	
1 3	
2 3	
3 3	
4 3	
5 3	
1 4	
1 5	
2 1	
5 5	

Задача G. Чаепитие

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

В одном из отделов крупной организации работает n человек. Как практически все сотрудники этой организации, они любят пить чай в перерывах между работой. При этом они достаточно дисциплинированы и делают в день ровно один перерыв, во время которого пьют чай. Для того, чтобы этот перерыв был максимально приятным, каждый из сотрудников этого отдела обязательно пьет чай одного из своих любимых сортов. В разные дни сотрудник может пить чай разных сортов. Для удобства пронумеруем сорта чая числами от 1 до m .

Недавно сотрудники отдела купили себе большой набор чайных пакетиков, который содержит a_1 пакетиков чая сорта номер 1, a_2 пакетиков чая сорта номер 2, ..., a_m пакетиков чая сорта номер m . Теперь они хотят знать, на какое максимальное число дней им может хватить купленного набора так, чтобы в каждый из дней каждому из сотрудников доставался пакетик чая одного из его любимых сортов.

Каждый сотрудник отдела пьет в день ровно одну чашку чая, которую заваривает из одного пакетика. При этом пакетики чая не завариваются повторно.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 50$). Вторая строка содержит m целых чисел $a_1, \dots, a_m$ ($1 \leq a_i \leq 10^6$ для всех i от 1 до m).

Далее следуют n строк — i -я из этих строк описывает любимые сорта i -го сотрудника отдела и имеет следующий формат: сначала следует положительное число k_i — количество любимых сортов чая этого сотрудника, а затем идут k_i различных чисел от 1 до m — номера этих сортов.

Формат выходных данных

Выведите одно целое число — искомое максимальное количество дней.

Примеры

tea.in	tea.out
2 3 3 2 1 2 1 2 2 1 3	3

Задача Н. Химия!!!

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

Вася и Сережа играют в следующую игру. В некоторых клетках клетчатого листка Сережа рисует один из символов 'H', 'O', 'N' или 'C', после чего Вася должен провести между некоторыми находящимися в соседних клетках символами линии так, чтобы получилось корректное изображение химической молекулы. К сожалению, Сережа любит рисовать много символов, и Вася не может сразу определить, возможно ли вообще нарисовать линии нужным способом. Помогите ему написать программу, которая даст ответ на этот вопрос.

В этой задаче проведенные между символами химических элементов линии будем считать корректным изображением молекулы, если они удовлетворяют следующим условиям:

- каждая линия соединяет символы, нарисованные в соседних (по стороне) клетках,
- между каждой парой символов проведено не более одной линии,
- от каждого элемента отходит ровно столько линий, какова валентность этого элемента (1 для H, 2 для O, 3 для N, 4 для C),
- пустые клетки ни с чем не соединены, и
- хотя бы в одной клетке нарисован какой-то символ.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m ($1 \leq n, m \leq 50$) — размеры листочка, на котором рисует Сережа. Далее следуют n строк по m символов в каждой, задающих конфигурацию химических элементов, которую нарисовал Сережа; пустые клетки задаются символом '.'.

Формат выходных данных

В выходной файл выведите одно слово: 'Valid', если линии провести требуемым образом можно, и 'Invalid', если нельзя.

Примеры

molecule.in	molecule.out
3 4 HOH. NCOH OO..	Valid
3 4 HOH. NCOH OONH	Invalid

Задача I. Perspective

Имя входного файла: `perspective.in`
Имя выходного файла: `perspective.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

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.

Примеры

perspective.in	perspective.out
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