

Задача А. Максимальный поток

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

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

стандартный ввод	стандартный вывод
4 5 1 2 1 1 3 2 3 2 1 2 4 2 3 4 1	3 1 2 1 2 1
4 5 1 2 2 1 3 2 2 3 1 2 4 2 3 4 2	4 2 2 0 2 2
4 3 1 2 1 1 3 1 1 4 4	4 0 0 4

Задача В. Максимальный поток минимальной стоимости

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задан ориентированный граф, каждое ребро которого обладает пропускной способностью и стоимостью. Найдите максимальный поток минимальной стоимости 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 ($2 \leq n \leq 100$, $0 \leq m \leq 1000$). Следующие m строк содержат по четыре целых числа: номера вершин, которые соединяет соответствующее ребро графа, его пропускную способность и его стоимость. Пропускные способности и стоимости не превосходят 10^5 .

Формат выходных данных

В выходной файл выведите одно число — цену максимального потока минимальной стоимости из вершины с номером 1 в вершину с номером n . Ответ не превышает $2^{63} - 1$. Гарантируется, что в графе нет циклов отрицательной стоимости.

Примеры

стандартный ввод	стандартный вывод
4 5 1 2 1 2 1 3 2 2 3 2 1 1 2 4 2 1 3 4 2 3	12

Замечание

В этой задаче достаточно несколько раз пустить Форд-Беллмана...

Задача С. Задача о назначениях

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

Дана целочисленная матрица C размера $n \times n$. Требуется выбрать n ячеек так, чтобы в каждой строке и каждом столбце была выбрана ровно одна ячейка, а сумма значений в выбранных ячейках была минимальна.

Формат входных данных

Первая строка входного файла содержит n ($2 \leq n \leq 300$). Каждая из последующих n строк содержит по n чисел: C_{ij} . Все значения во входном файле неотрицательны и не превосходят 10^6 .

Формат выходных данных

В первую строку выходного файла выведите одно число — искомая минимизируемая величина. Далее выведите n строк по два числа в каждой — номер строки и столбца клетки, участвующей в оптимальном назначении.

Пары чисел можно выводить в произвольном порядке.

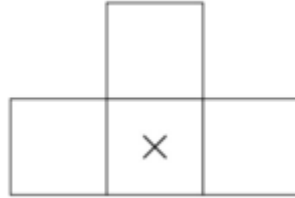
Примеры

<code>stdin</code>	<code>stdout</code>
3	3
3 2 1	2 1
1 3 2	3 2
2 1 3	1 3

Задача D. T - Покрытие

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Если вы когда-нибудь играли в Тетрис, вы должны знать, что одна из фигур выглядит так:



Мы будем называть эту фигуру T-тетрамино; тетрамино — это просто модное слово для связной геометрической фигуры, состоящей из 4 клеток. Клетка, отмеченная X, будет называться центральной клеткой.

Манка рисует прямоугольную сетку с m рядами и n столбцами и записывает некоторое число в каждую клетку. Ряды таблицы пронумерованы от 0 до $m - 1$, и столбцы от 0 до $n - 1$. Она также отмечает некоторые клетки как особые, например, окрашивая их в красный цвет. После этого она просит Нику, свою подругу, расположить несколько T-тетрамино на сетке таким образом, чтобы выполнялись следующие условия:

1. Количество T-тетрамино должно быть таким же, как и количество особых клеток.
2. Для каждого T-тетрамино, его центральная клетка должна лежать на особой клетке.
3. Ни одна пара T-тетрамино не должна перекрываться.
4. Все T-тетрамино должны полностью лежать в нарисованной сетке.

Обратите внимание, что существует 4 возможных расположения для каждого T-тетрамино (\neg , T, \vdash and \perp).

Если условия не могут быть выполнены, Ника должна ответить No; если они выполнимы, она должна найти такое расположение T-тетрамино, чтобы максимизировать сумму чисел в клетках, покрытых T-тетрамино. В этом случае она должна сообщить Манке максимальную сумму покрытых T-тетрамино клеток.

Напишите программу, чтобы помочь Нике решить эту задачу.

Формат входных данных

Первая строка входных данных содержит два целых числа m и n ($1 \leq mn \leq 10^6$) — высоту и ширину таблицы. Следующие m строк содержат по n целых чисел в интервале $[0, 1000]$. j -е целое число в i -й строке соответствует числу, записанному в j -й клетке i -го ряда сетки.

Следующая строка содержит число k ($1 \leq k \leq \min(nm, 1000)$) — количество особых клеток в таблице. Каждая из последующих строк содержит по два числа r_i и c_i , $r_i \in \{0, \dots, m - 1\}$ и $c_i \in \{0, \dots, n - 1\}$ — номер строки и столбца i -й особой клетки, соответственно.

Гарантируется, что все координаты особых клеток уникальны.

Формат выходных данных

Выведите максимально возможную сумму чисел, покрытых T-тетрамино клеток, либо No, если не существует способа расставить их.

Примеры

стандартный ввод	стандартный вывод
5 6 7 3 8 1 0 9 4 6 2 5 8 3 1 9 7 3 9 5 2 6 8 4 5 7 3 8 2 7 3 6 3 1 1 2 2 3 4	67
5 6 7 3 8 1 0 9 4 6 2 5 8 3 1 9 7 3 9 5 2 6 8 4 5 7 3 8 2 7 3 6 3 1 1 2 2 3 3	No

Задача Е. Приготовление еды

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Степан и Сергей поступили в университет. Началась взрослая жизнь, а значит, теперь им нужно готовить еду самостоятельно.

Друзья умеют готовить n различных блюд. Купив все необходимые продукты, ребята поняли, что до следующего похода в магазин они хотят приготовить i -е блюдо ровно a_i раз.

Каждый день Сергей и Степан выбирают два блюда i и j и готовят их, это занимает $c_{i,j}$ единиц времени. При этом возможна ситуация, когда $i = j$, тогда в этот день i -е блюдо оказывается приготовлено дважды.

Поскольку ребята довольно ленивые, они хотят минимизировать суммарное время приготовления блюд за все дни до следующего похода в магазин. Помогите им в этом!

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 10$) — количество блюд, которые ребята умеют готовить.

Вторая строка содержит n положительных целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 50$) — для каждого блюда указано, сколько раз его нужно приготовить.

В каждой из следующих n строк записаны n целых чисел $c_{i,j}$ ($1 \leq c_{i,j} \leq 100$), j -е число в i -й строке обозначает время приготовления пары блюд i и j . Гарантируется, что $c_{i,j} = c_{j,i}$.

Формат выходных данных

Выведите одно целое число: минимальное суммарное время приготовления блюд либо -1 , если невозможно составить план приготовления блюд так, чтобы i -е было приготовлено ровно a_i раз.

Примеры

стандартный ввод	стандартный вывод
3 2 2 2 1 4 3 4 4 5 3 5 6	10
2 2 39 23 9 9 23	-1
1 2 100	100

Замечание

В первом примере оптимально приготовить следующие пары блюд: $(1, 3)$, $(1, 3)$, $(2, 2)$.

Задача F. Банах

Имя входного файла: `banach.in`
Имя выходного файла: `banach.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Стефан, Дэвид и Феликс готовят соревнование по правилам ICPC. Стефан предлагают следующую задачу:

Дано N точек на плоскости, а также N векторов. Найдите однозначное соответствие между векторами и точками так, что если передвинуть каждую точку на соответствующий вектор, то расстояние между **любой** парой точек не уменьшится.

Дэвид справился с решением этой задачи довольно быстро и утверждает, что эта задача слишком проста для этого соревнования. Для того, чтобы убедить его, Феликс предложил следующее усложнение: среди всех возможных решений, найдите то, которое максимизирует сумму квадратов всех результирующих попарных расстояний.

Дэвиду все еще кажется, что задача слишком простая, а вам?

Формат входных данных

Первая строка входного файла содержит число N , количество точек и векторов ($1 \leq N \leq 500$).

Следующие N строк описывают точки. Каждая из них содержит два числа px_i и py_i ($0 \leq |px_i|, |py_i| \leq 10\,000$).

Затем следуют N строк с описанием векторов. Каждый вектор задан двумя числами vx_i и vy_i ($0 \leq |vx_i|, |vy_i| \leq 10\,000$).

Формат выходных данных

Если способ сопоставить точки и вектора таким образом, чтобы все попарные расстояния не уменьшились существует, выведите "Yes" в первой строке выходного файла. На следующей строке, выведите N различных чисел от 1 до N , i -е из которых равно номеру вектора, сопоставленного i -й точке.

Не забудьте выбрать ответ с максимально возможной суммой квадратов всех результирующих попарных расстояний. Если существует несколько возможных решений, выведите любое из них.

Если требование выполнить невозможно, выведите "No" в единственной строке выходного файла.

Примеры

<code>banach.in</code>	<code>banach.out</code>
2 0 0 1 0 2 0 -2 0	Yes 2 1
2 2 2 2 2 -1 -1 -1 -1	Yes 1 2

Замечание

В первом примере, существует только два возможных способа сопоставить векторы точкам. В обоих сопоставлениях, "1 2" и "2 1", расстояние между каждой парой точек не уменьшается. В первом случае сумма квадратов расстояний между всеми парами точек равна 9, но во втором случае она равна 25. Поэтому единственным правильным решением является "2 1".

Во втором примере, опять же, только два случая установить соответствие. В обоих случаях, расстояние между всеми парами точек не уменьшается, и сумма квадратов попарных расстояний равна 0. Поэтому, оба ответа корректны.