

Задача А. 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 В. 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

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

Задача С. Автоматное программирование

Имя входного файла: `schedule.in`
Имя выходного файла: `schedule.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

В один замечательный день в компанию «Х» завезли k автоматов. И не простых автоматов, а автоматов-программистов! Это был последний неудачный шаг перед переходом на андроидов-программистов, но это уже совсем другая история.

В компании сейчас n задач, для каждой из которых известно время начала ее выполнения s_i , длительность ее выполнения t_i и прибыль компании от ее завершения c_i . Любой автомат может выполнять любую задачу, ровно одну в один момент времени. Если автомат начал выполнять задачу, то он занят все моменты времени с s_i по $s_i + t_i - 1$ включительно и не может переключиться на другую задачу.

Вам требуется выбрать набор задач, которые можно выполнить с помощью этих k автоматов и который принесет максимальную суммарную прибыль.

Формат входных данных

В первой строке записаны два целых числа n и k ($1 \leq n \leq 1000$, $1 \leq k \leq 50$) — количество задач и количество автоматов, соответственно.

В следующих n строках через пробелы записаны тройки целых чисел s_i, t_i, c_i ($1 \leq s_i, t_i \leq 10^9$, $1 \leq c_i \leq 10^6$), s_i — время начала выполнения i -го задания, t_i — длительность i -го задания, а c_i — прибыль от его выполнения.

Формат выходных данных

Выведите n целых чисел $x_1, x_2, \dots, x_n$. Число x_i должно быть равно 1, если задачу i следует выполнить, и 0 в противном случае.

Если оптимальных решений несколько, то выведите любое из них.

Примеры

schedule.in	schedule.out
3 1 2 7 5 1 3 3 4 1 3	0 1 1
5 2 1 5 4 1 4 5 1 3 2 4 1 2 5 6 1	1 1 0 0 1

Задача D. Аарельские горы

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

Вы — могучий волшебник, путешествующий по Аарельскому королевству. Ваш путь лежит через крутые холмы и горы. Горная цепочка может быть представлена, как последовательность n регионов; каждый регион имеет свою *высоту* h_i .

К сожалению, у вас нет никакого скалолазного оборудования (и, честно говоря, никакого умения его использовать), и ваши заклинания полета требуют перезарядки. Однако, вы сможете пересечь эту область, если высоты регионов будут образовывать неубывающую последовательность, то есть $h_i \leq h_{i+1}$ для всех i от 1 до $(n - 1)$ (вы путешествуете справа налево, и падения с высокой высоты для вас опасности не представляют).

Единственное заклинание, которое может вам помочь, это заклинание Землетрясение. Это заклинание влияет на несколько соседних регионов и позволяет вам изменить их высоты. Вы знаете m видов Землетрясения; каждое заклинание имеет свою длину l_i (количество соседних регионов, затронутых этих заклинанием), свою энергетическую стоимость c_i и либо увеличивает, либо уменьшает высоты всех затронутых регионов (одно заклинание может либо повышать, либо понижать регионы, но не и то, и то). Каждое заклинание может быть применено к любому отрезку длины l_i , полностью лежащему внутри области из n регионов, и может быть применено сколько угодно раз (но его энергетическую стоимость придется платить при каждом применении). Разрешено, чтобы какие-то регионы имеют отрицательную высоту после применения заклинания.

Определите, возможно ли изменить ландшафт так, что вы сможете пересечь заданную область, и, если это возможно, найдите минимальную суммарную энергию, которую придется потратить на применение заклинаний.

Формат входных данных

Первая строка содержит два числа n и m , разделенных пробелами ($1 \leq n, m \leq 200$).

Вторая строка содержит n чисел h_i — изначальную высоту регионов ($0 \leq h_i \leq 10^6$).

Следующих m строчек описывают виды заклинаний Землетрясение, которые вы знаете, i -я из этих строк содержит описание заклинания в формате " $t_i l_i c_i$ "; где t_i это символ "+" если заклинание позволяет увеличить высоту на 1, или "-" если оно позволяет уменьшить высоту на 1; l_i и c_i — длина и энергетическая стоимость заклинания, соответственно ($1 \leq l_i \leq n, 1 \leq c_i \leq 10^6$).

Формат выходных данных

Если вы можете пройти через эту область, выведите минимальную возможную суммарную энергию, потраченную на заклинания, иначе, выведите -1 .

Примеры

стандартный ввод	стандартный вывод
3 2 3 2 1 + 1 1 - 1 1	2
3 1 3 2 1 + 2 1	-1

Задача Е. Транспортировка товаров

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Вдоль трассы с односторонним движением расположены n городов. Города пронумерованы числами от 1 до n в порядке проезда вдоль трассы.

В i -м городе было произведено p_i единиц товара и может быть продано не более чем s_i единиц товара.

Для каждой пары городов i и j , таких что $1 \leq i < j \leq n$, можно не более одного раза перевезти не более чем c единиц товара из города i в город j . Заметьте, что товар можно перевозить только из города с меньшим номером в город с большим номером. **Перевозить товары между городами можно в любом порядке.**

Определите, какое максимальное количество произведённого товара суммарно может быть продано во всех городах после некоторой последовательности перевозок.

Формат входных данных

Первая строка входных данных содержит два числа n и c ($1 \leq n \leq 10\,000$, $0 \leq c \leq 10^9$) — количество городов и максимальное количество товаров, которое можно перевезти между городами за один раз.

Во второй строке записаны n чисел p_i ($0 \leq p_i \leq 10^9$) — количество единиц товара, произведённого в каждом городе.

В третьей строке записаны n чисел s_i ($0 \leq s_i \leq 10^9$) — количество единиц товара, которое может быть продано в каждом городе.

Формат выходных данных

Выведите одно число — максимальное количество товара, которое может быть продано во всех городах после некоторой последовательности перевозок.

Примеры

стандартный ввод	стандартный вывод
3 0 1 2 3 3 2 1	4
5 1 7 4 2 1 0 1 2 3 4 5	12
4 3 13 10 7 4 4 7 10 13	34

Замечание

Задача с финала Intel Code Challenge, прошедшего 08.10.2016 на сайте codeforces.com.

Задача F. 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 G. 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.