

## Условие

Разработать жадный алгоритм решения задачи, определяемой своим вариантом. Доказать его корректность, оценить скорость и объём затрачиваемой оперативной памяти.

Разработать программу на языке C или C++, реализующую построенный алгоритм. Формат входных и выходных данных описан в варианте задания.

## Вариант 4. Откорм бычков

Бычкам дают пищевые добавки, чтобы ускорить их рост. Каждая добавка содержит некоторые из  $N$  действующих веществ. Соотношения количеств веществ в добавках могут отличаться. Воздействие добавки определяется как

$$c_1 a_1 + c_2 a_2 + \dots + c_N a_N,$$

где  $a_i$  — количество  $i$ -го вещества в добавке,  $c_i$  — неизвестный коэффициент, связанный с веществом и не зависящий от добавки. Чтобы найти неизвестные коэффициенты  $c_i$ , Биолог может измерить воздействие любой добавки, используя один её мешок. Известна цена мешка каждой из  $M$  ( $N \leq M$ ) различных добавок. Нужно помочь Биологу подобрать самый дешевый набор добавок, позволяющий найти коэффициенты  $c_i$ . Возможно, соотношения веществ в добавках таковы, что определить коэффициенты нельзя.

### Входные данные

В первой строке текста — целые числа  $M$  и  $N$ ; в каждой из следующих  $M$  строк записаны  $N$  чисел, задающих соотношение количеств веществ в ней, а за ними — цена мешка добавки. Порядок веществ во всех описаниях добавок один и тот же, все числа — неотрицательные целые не больше 50.

### Выходные данные

Если определить коэффициенты невозможно: -1, иначе набор добавок (и их номеров по порядку во входных данных). Если вариантов несколько, вывести какой-либо из них.

## Описание алгоритма:

В задаче требуется составить систему уравнений имеющую единственный корень. Из линейной алгебры известно, что единственное решение существует только в случае когда определитель матрицы не равен нулю. А это в свою очередь означает что необходимо найти  $N$  линейно независимых строк.

Очевидно что для решения данной задачи не нужно перебирать все возможные комбинации строк необходимо брать строки с минимальной ценой. Поэтому перед началом алгоритма отсортируем строки по возрастанию цены с помощью пирамидальной сортировки. После этого создаем массив строк и забиваем его первыми  $N$  элементами из отсортированного массива. Далее запускаем цикл до тех пор пока остаются необработанные строки. В цикле запускаем функцию поиска "плохой строки" (на самом деле это алгоритм Гаусса, но если строки матрицы линейно зависимы то эта функция возвращает индекс строки которую необходимо заменить или 0 в случае успеха).

Реализация данной функции: для каждого столбца находим первый отличный от нуля элемент и перемещаем эту строку наверх. Затем из всех строк имеющих ненулевые элементы в текущем столбце вычитаем найденную строку умноженную на коэффициент таким образом чтобы в столбце остался только один элемент не равный нулю. Если же в столбце не нашлось элементов отличных от нуля, то проходим по матрице и выбираем строку с самой большой ценой (при этом строки перемещенные на предыдущих этапах наверх не рассматриваются) и возвращаем номер этой строки смещенный на 1. Если в процессе выполнения алгоритма нулевых столбцов не возникло, то возвращаем 0.

Если вышеописанная функция вернула значение 0, то выходим из цикла, иначе заменяем строку по полученному индексу смещенному на 1 влево и увеличиваем счетчик использованных строк. Продолжаем выполнение цикла.

После выхода из цикла проверяем сколько строк было использовано: если использовано строк меньше  $M$ , то мы нашли ответ и можно выводить номера строк в исходной последовательности предварительно отсортировав их. Иначе мы использовали все строки но так и не нашли  $N$  линейно независимых, поэтому выводим -1.

## **Сложность:**

Сложность алгоритма Гаусса -  $O(n^3)$ . В худшем случае мы запускаем его  $M - N$  раз. Итого сложность  $O((M - N)N^3)$

## **Дневник отладки**

Возникли проблемы с выводом ответа. Изначально выводилась сумма которую необходимо заплатить, но такое решение получило  $WA\#2$  после чего программа была изменена и в качестве ответа выдавались номера строк которые нужно взять. Но такое решение получило  $WA\#3$  после чего номера строк дополнительно были отсортированы. После этого программа получила  $OK$

## **Выводы**

Решение задач при помощи жадных алгоритмов не представляет особой сложности. В принципе жадные алгоритмы это разновидность динамического программирования, только вместо разбиения на несколько подзадач задача сводится сразу же к задаче меньшей размерности за счет выбора наилучшего варианта на текущем шаге. Однако такой подход не всегда применим. Как и в динамическом программировании это должны быть задачи оптимизации или что-нибудь подобное, а также необходимо чтобы ответ всегда существовал иначе не всегда корректно будет выбирать наилучший вариант на текущем шаге.