

Лабораторная работа № 8 по курсу Дискретный Анализ. Жадные алгоритмы

Выполнил студент группы М8О-307Б-21 МАИ *Друхольский Александр*.

Условие

Задача:

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 $c_1a_1 + c_2a_2 + \dots + c_Na_N$, 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 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.

Метод решения

Стоит отметить, что задача по своей сути сходится к задаче о возможности решения системы линейных уравнений. Проверять на возможность решения мы будем с помощью метода Гаусса, приводя матрицу к ступенчатому виду. Перефразируем задачу следующим образом. Пусть дана система уравнений, у каждого из которых есть цена. Нужно узнать, возможно ли решить систему и если да, то используя наиболее дешёвые уравнения. Применим метод жадного алгоритма: каждый раз будем в первую очередь рассматривать новое уравнение с минимальной ценой (то есть будем менять строки матрицы в порядке убывания цены).

Фиксируем i столбец, выбираем очередную строку с минимальным значением цены без нуля в i -ом столбце, меняем местами i строку с выбранной строкой. Далее приводим матрицу к ступенчатому виду для этого шага. Идём к следующему столбцу и повторяем алгоритм. Если на каком-то из шагов не получается выделить ненулевую строку, то значит в матрице будет ноль на диагонали, а следовательно такую систему решить нельзя. Итого сложность алгоритма $O(n * n * m)$, так как нам надо будет пройти по каждому члену матрицы для построения ступенчатого вида.

Описание программы

Основные моменты:

1. `vector < vector < double >> data(m, vector < double > (n + 2, 0));` - матрица, которую будем проверять. В ней дополнительно сохраняем номер уравнения для вывода в отсортированном порядке. Используем тип данных `double`, так как при построении ступенчатой матрицы на промежуточных этапах могут быть нецелые значения.
2. Основное тело программы - цикл, которые пробегается по каждому столбцу. В этом цикле - поиск строки с минимальной ценой и приведение матрицы к ступенчатому виду (преобразования строк) для текущего шага.

Дневник отладки

WA - 3 Ошибка с выводом

WA - 4 Ошибка в логике алгоритма в подсчёте ступенчатой матрицы.

Тест производительности

Для замера времени будем увеличивать n и m в 10 раз (хотя по условию задачи они меньше 50). Для удобства возьмём $n = m$ и исследуем на сложность $O(n^3)$

№	Число $N=M$	Время, с
1	10	0.000027
2	100	0.003949
3	1000	1.99866

Как мы видим, время выполнения увеличивается на несколько порядков. Тесты при больших n особого смысла не имеют (слишком много времени). Алгоритм эффективен в рамках чисел, которые устанавливает условие.

Выводы

Я решил задачу методом жадного алгоритма. Разобрался в отличиях между динамическим программированием и жадными алгоритмами.