

Лабораторная работа №8 по курсу дискретного анализа: жадные алгоритмы.

Выполнил студент группы М80-308Б-20 Морозов Артем Борисович.

Условие:

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, использовав один её мешок. Известна цена мешка каждой из M ($M \geq N$) 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: -1 если определить коэффициенты невозможно, иначе набор добавок (и их номеров по порядку во входных данных). Если вариантов несколько, вывести какой-либо из них.

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

Метод решения

Метод решения данной задачи основывается на такой идее, как жадные алгоритмы. Суть жадных алгоритмов в том, что мы на каждом шаге берем минимальный, то есть оптимальный для нас ответ, чтобы в итоге суммарным ответом на задачу была сумма минимальных ответов на i -тых итерациях. То есть суммарный ответ так же был минимальный. Что же касается задачи с бычками, то алгоритм довольно прост: чтобы решить эту задачу, нам нужно найти N линейно независимых строк в исходных данных, то есть привести нашу матрицу к ступенчатому виду методом Гаусса. Если это сделать не удалось (мы получили 0 в первом индексе), то возвращаем -1. Иначе возвращаем отсортированный список номеров строк, в которых содержатся минимальные добавки. Сложность алгоритма – $O(M \cdot N^3)$, так как изначально мы проходимся циклом от 0 до N , в цикле от 1 до M мы ищем минимальную стоимость и нужный столбец, и за $O(N^2)$ мы суммарно вычитаем строки методом Гаусса.

Описание программы

Программа сделана в одном файле. Все действие происходит в функции `main` и во вспомогательной функции `Find`, в которой и написан алгоритм решения задачи.

Дневник отладки

В этой лабораторной довольно послушной список отладки. С самого начала у меня была проблема с 1- и 0- индексацией на моей локальной машине, в связи с чем я долго не мог получать правильный ответ при правильной реализации алгоритма. После этого у меня ушло много попыток отправить задачу на чекер, но он выдавал `runtime error`, что меня очень смущало, учитывая корректную работу программы на локальной машине. Дело было в том, что я сначала объявлял двумерный вектор, а потом только инициализировал его размеры. Оказывается, компилятор C++ на это не ругается, а сам делает нужные оптимизации, поэтому задача у меня решалась верно. Однако чекер такое не пропускал.

Тест производительности

Сравним время выполнения задачи на разном объеме данных:

1 тест – $M = 10$, $N = 10$

TIME: 3 milliseconds

2 тест – $M = 100$, $N = 100$

TIME: 37 milliseconds

3 тест – $M = 1000$, $N = 1000$

2955 milliseconds

4 тест – $M = 50$, $N = 500$

TIME: 164 milliseconds

5 тест – $M = 500$, $N = 50$

TIME: 98 milliseconds

Таким образом, мы видим, что сложность алгоритма по большей части зависит от N , нежели от M , как я и говорил.

Недочёты

Недочётов в программе обнаружено не было, однако стоит упомянуть, что программа работает только при условии корректного ввода, так как была разработана исключительно в учебных целях. Любой неправильный ввод может убить работоспособность моей программы.

Выводы

Данная лабораторная работа помогла мне лучше осознать такую идею, как жадные алгоритмы. Хотя я и раньше слышал про подобное, именно сейчас я непосредственно воспользовался этой концепцией и решил весьма интересную задачу, которая, помимо улучшения моих навыков программирования, заставила меня вспомнить первый курс линала.

