

Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №8 по курсу «Дискретный анализ»

Студент: А. С. Бычков
Преподаватель: Н. К. Макаров
Группа: М8О-301Б
Дата:
Оценка:
Подпись:

Москва, 2024

Лабораторная работа №8

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 $M \leq N$ 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.

Формат ввода: 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.

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

1 Описание

Во-первых, эту задачу можно преобразовать в математический вид: У нас есть система из M уравнений:

$$a_{11}c_1 + a_{12}c_2 + \dots + a_{1n}c_n$$

$$a_{21}c_1 + a_{22}c_2 + \dots + a_{2n}c_n$$

...

$$a_{m1}c_1 + a_{m2}c_2 + \dots + a_{mn}c_n$$

Очевидно, если $M < N$, она имеет бесконечное множество решений. Если же $M \geq N$, то эта система будет иметь единственное решение тогда и только тогда, когда ранг матрицы, составленной из N уравнений будет равен N .

Жадность алгоритма заключается в следующем:

1. Возьмем строку с минимальной стоимостью
2. Найдем строку, которая будет ЛНЗ с первой, причем ее стоимость будет также минимальной из всех ЛНЗ строк
3. Продолжим так до тех пор, пока не найдем все N ЛНЗ строк.

ЛНЗ будем проверять путем приведения строк в ступенчатому виду.

2 Исходный код

```
1 #include <vector>
2 #include <iostream>
3 #include <optional>
4 #include <algorithm>
5
6
7 std::optional<int> find_row_index(const std::vector<std::vector<double>>& matrix,
8     const std::vector<int>& prices, int column_index) {
9     int min_cost = 1e9;
10    std::optional<int> min_index;
11
12    for (int i = column_index; i < matrix.size(); ++i) {
13        if (matrix[i][column_index] != 0 && prices[i] < min_cost) {
14            min_index = i;
15            min_cost = prices[i];
16        }
17    }
18    return min_index;
19 }
20
21 void do_substraction(std::vector<std::vector<double>>& matrix, int base) {
22     for (int i = base + 1; i < matrix.size(); ++i) {
23         double coef = matrix[i][base] / matrix[base][base];
24         for (int j = base; j < matrix[0].size() - 1; ++j) {
25             matrix[i][j] -= matrix[base][j] * coef;
26         }
27     }
28 }
29
30 int main() {
31     int n, m;
32     std::cin >> m >> n;
33     std::vector<std::vector<double>> matrix(m, std::vector<double>(n + 1));
34     std::vector<int> prices(m);
35     int ans = 0;
36     std::vector<int> lines;
37
38     for (int i = 0; i < m; ++i) {
39         for (int j = 0; j < n; ++j) {
40             std::cin >> matrix[i][j];
41         }
42         matrix[i][n] = i;
43         std::cin >> prices[i];
44     }
45
46     if (m < n) {
```

```

47     std::cout << -1;
48     return 0;
49 }
50
51 for (int i = 0; i < n; ++i) {
52     auto min_index = find_row_index(matrix, prices, i);
53     if (!min_index) {
54         std::cout << -1;
55         return 0;
56     }
57     std::swap(matrix[i], matrix[min_index.value()]);
58     std::swap(prices[i], prices[min_index.value()]);
59     ans += prices[i];
60     lines.push_back(matrix[i][n]);
61     do_substraction(matrix, i);
62 }
63
64 std::sort(lines.begin(), lines.end());
65 for (const auto& el : lines) {
66     std::cout << el + 1 << ' ';
67 }
68 }

```

3 Консоль

```
g++ greedy.cpp  
./a.out  
3 3  
1 0 2 3  
1 0 2 4  
2 0 1 2  
-1%
```

4 Выводы

Благодаря данной лабораторной работе я познакомился с жадными алгоритмами.

Список литературы

- [1] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. *Алгоритмы: построение и анализ, 2-е издание.* — Издательский дом «Вильямс», 2007. Перевод с английского: И. В. Красиков, Н. А. Орехова, В. Н. Романов. — 1296 с. (ISBN 5-8459-0857-4 (рус.))