Отчет по лабораторной работе дискретного анализа “жадные алгоритмы”

Преподаватель:
Макаров Никита
Студент выполнил:
Мудров П.Ф
М8О-303Б-22

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 $c_1a_1 + c_2a_2 + \cdots + c_{NaN}$, 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 ($\mathrm{M} \leqslant \mathrm{N}$) 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.

Метод решения

1. **Функция для выбора строки (SelectRow):**
   - Принимает текущий столбец.
   - Возвращает индекс строки с минимальным приоритетом (или -1, если строки нет).
2. **Функция для обработки строк ниже текущей (Eliminate Below):**
   - Устраняет значения в текущем столбце для всех строк ниже текущей.
3. **Основной цикл обработки:**

- Для каждого столбца:
    1. Вызывается функция SelectRow для поиска подходящей строки.
    2. Производится перестановка строк.
    3. Выполняется операция “устранения” с помощью Eliminate Below.
- Фиксируются индексы выбранных строк.

4. **Сортировка и вывод результата:**

- Индексы выбранных строк сортируются и выводятся.

Асимптотика

- **Временная сложность:** O(m^2 * n) .
- **Пространственная сложность:** O(m* n) .

Исходный код

```
#include <iostream>

#include <vector>

#include <limits>

#include <algorithm>


int SelectRow(std::vector<std::vector<double>>& matrix, int
    columnIndex, int m, int n) {

    double lowestPrice = std::numeric_limits<double>::max();

    int selectedRow = -1;


    for (int row = columnIndex; row < m; ++row) {

        if ((matrix[row][columnIndex] != 0.0) &&
    (matrix[row][n] < lowestPrice)) {

            selectedRow = row;

            lowestPrice = matrix[row][n];

        }

    }


    return selectedRow;
```

```
}



void EliminateBelow(std::vector<std::vector<double>>& matrix,
    int pivotRow,int m, int n) {
    for (int currentRow = pivotRow + 1; currentRow < m;
    ++currentRow) {
        if (matrix[pivotRow][pivotRow] == 0.0) continue;
        double factor = matrix[currentRow][pivotRow] /
    matrix[pivotRow][pivotRow];
        for (int col = pivotRow; col < n; ++col) {
            matrix[currentRow][col] -= factor *
    matrix[pivotRow][col];
        }
    }
}


int main(){
    int m,n;
    std::cin >> m >> n;
    std::vector<std::vector<double>>
    matrix(m,std::vector<double>(n+2));
    std::vector<int> ans;
    for(int i=0 ;i<m; i++){
        for(int j=0;j<n+1;j++){
            std::cin >> matrix[i][j];
        }
        matrix[i][n+1] = i;
    }


        for (int i = 0; i < n; ++i) {
        int index = SelectRow(matrix, i,m,n);
```

```
            if (index == -1) {
                std::cout << "-1" << std::endl;
                return 0;
            }

            std::swap(matrix[i], matrix[index]);
            ans.push_back(matrix[i][n + 1]);
            EliminateBelow(matrix, i,m,n);
        }

        std::sort(ans.begin(), ans.end());
        for (int i = 0; i < ans.size(); ++i) {
            std::cout << ans[i] + 1;
            if (i == ans.size() - 1) {
                std::cout << std::endl;
            } else {
                std::cout << " ";
            }
        }

        return 0;
    }
```

Тесты

1:
3 3
1 2 3 4
2 3 1 6
3 1 2 7

1 2 3

2 :
3 3
0 0 0 1
0 0 0 2
0 0 0 3

-1

3:
3 3
0 1 0 3
1 0 0 4
0 0 1 5

2 1 3

4:
4 3
1 0 0 1
0 1 0 2
0 0 1 3
1 1 1 4

1 2 3

5:
5 4
2 1 3 4 5
1 4 2 1 6
3 2 1 3 7
4 3 2 4 8
1 1 1 1 9

1 2 3 4 5

Вывод:

В данной лабораторной работе я реализовал алгоритм для обработки матрицы с целью упорядочивания строк в соответствии с заданными условиями. Задача заключалась в реализации методов выбора строки с минимальным “приоритетом”, перестановки строк матрицы и приведения её к треугольной форме путем вычитания пропорциональных строк. Я использовал стандартные методы работы с двумерными массивами и основные алгоритмические подходы, включая жадный поиск и сортировку.

В процессе выполнения я изучил основы работы с матрицами в программировании, углубился в применение алгоритмов выбора и их оптимизацию. Основной вызов заключался в обработке

исключительных случаев, таких как строки, полностью состоящие из нулей, или ситуации, где невозможен выбор подходящей строки. Для таких случаев я предусмотрел обработку ошибок и завершение программы с выводом соответствующего сообщения.

Алгоритм успешно справился с различными тестовыми примерами, включая базовые случаи (упорядоченные матрицы), требующие перестановки строки, а также случаи с невозможностью корректного упорядочивания. Были протестированы матрицы разного размера, включая те, где число строк превышает число столбцов. В результате я смог убедиться в корректности работы программы и ее способности решать задачу в рамках заданных ограничений.

В целом, работа над данной задачей помогла мне закрепить знания о работе с двумерными структурами данных, алгоритмах выбора и сортировки, а также научила уделять внимание деталям, таким как обработка граничных случаев и оптимизация ресурсоемких операций.