

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

**«Московский Авиационный Институт»
(Национальный Исследовательский Университет)**

Факультет №8 «Компьютерные науки и прикладная математика»

Кафедра 805 «Прикладная математика»

**Курсовой проект
по курсу
«Архитектура ЭВМ, системное программное обеспечение»
2 семестр
Задание 7**

Автор работы:

студент 1 курса, гр. М8О-103Б-21

Фадеев Д.В.

Проверил:

Севастьянов В.С.

Дата сдачи:

Москва 2022 г

Содержание:

| | |
|---------------------------------|---|
| Задача. | 3 |
| Реализация. | 3 |
| Организация исходного кода..... | 4 |
| Описание реализации. | 4 |
| Выводы..... | 5 |

Задача.

Разработать программу на языке Си, позволяющую взаимодействовать с прямоугольными разреженными матрицами вещественного типа.

Разреженной матрицей называется такая матрица, в которой $>15\%$ элементов это нулевые значения.

Ввод матрицы происходит при построчном считывании чисел из файла, название которого вводит пользователь. Над созданной матрицей необходимо выполнить следующие действия. Добавление нового элемента, вывод матрицы в стандартном виде и в виде схемы, соответствующей представлению матрицы в памяти компьютера.

По заданию моего варианта мне требуется создать функции, транспонирующие матрицу по побочной диагонали и определяющие кососимметричность матрицы.

Реализация.

В моём задании представлена следующая схема реализации разреженной матрицы:

| | | | | | | |
|-----|--------------|---------------|----------|---------------|----------|-----|
| 0 | Номер строки | Номер столбца | Значение | Номер столбца | Значение | ... |
| ... | | | | | | |
| 0 | Номер строки | Номер столбца | Значение | ... | 0 | 0 |

То есть ненулевые элементы разреженной матрицы хранятся в листе, в котором начала строк разделены нулями, после чего пишется номер строки и стоят пары чисел: номер столбца и числа.

В коде для этой схемы реализована структура `vector`, в которой хранятся ссылки на предыдущие «ячейки» и значение самой клетки:

```
typedef struct vector {  
    double value;  
    struct vector *next;  
    struct vector *prev;  
}vector;
```

Организация исходного кода.

Программа разделена на три файла, задачи которых описаны в таблице.

| Файл | Задача |
|----------|---|
| vector.h | Заголовочный файл с описанным интерфейсом взаимодействия с матрицей |
| vector.c | Реализация интерфейса взаимодействия |
| main.c | Интерфейс взаимодействия пользователя и программы |

Описание реализации.

Начнём разбор реализации проекта с основных двух файлов: vector.h и vector.c.

В заголовочном файле vector.h описаны сигнатуры функций, которые потребуются для решения задания. Распишем их функционал:

1. create_vector() - выделение памяти под вектор.
2. create_from_file() - чтение матрицы из файла.
3. add_unit() - добавление числа в переданную матрицу в нужной строке и столбце.
4. add_unit_to_start() - добавление любого числа после другого.
5. find_row_start() - нахождение начала нужной строки в векторе.
6. print_normal() - вывод матрицы в классическом формате.
7. print_sch() - вывод в виде схемы.
8. trans() - транспонирование матрицы.
9. is_cososim() - определяет кососимметричность матрицы.

В vector.c эти все функции реализуются, а в main.c эти функции применяются к введённым пользователям данным.

Выводы.

Результатом курсовой работы стала программа, реализующая преобразования и действия над разряженными матрицами. Схема представления матрицы в памяти компьютера не заурядна и развивает абстрактное мышление.

Процесс создания алгоритмов добавления элементов, их взятия из матрицы был сложным. Вся проблема в запутанности схемы, приходится писать много итераторов и не путаться в ссылках на элементы, при добавлении нового. Поэтому чтобы при соединении пользовательского интерфейса и интерфейса взаимодействия с матрицей не было много сложного кода, были написаны высокоуровневные функции надстройки, которые соединяли воедино все итераторы.