

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ»
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)
ФАКУЛЬТЕТ №8 «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И
ПРИКЛАДНАЯ МАТЕМАТИКА»
КАФЕДРА 806 «ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И
ПРОГРАММИРОВАНИЕ»

Курсовой проект

На тему

«Разреженные матрицы»

Курс / Семестр:	1 / 2
Группа:	М8О-108Б-19
ФИО студента:	Горохов М.А.
ФИО преподавателя:	Поповкин А.В.
Подпись:	
Оценка:	
Дата сдачи:	
Дата проверки:	

Москва

2020

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ЦЕЛИ ПРОЕКТА.....	4
2. ПОСТАНОВКА ЗАДАЧИ.....	5
3. СТРУКТУРА ПРОЕКТА.....	6
4. ПРОГРАММНЫЙ КОД И ТЕСТОВЫЕ ДАННЫЕ.....	10
5. ЗАКЛЮЧЕНИЕ.....	15
6. ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ.....	16

ВВЕДЕНИЕ

В этом курсовом проекте я реализовал разреженные матрицы.

Разреженные матрицы - матрицы, имеющие не более половины ненулевых элементов. Они позволяют более компактно хранить данные в памяти ЭВМ.

Реализация была на массивах, с массивом индексации строк, индексов столбцов и сами значения.

1. ЦЕЛИ ПРОЕКТА

Изучить способы хранения и обработки матриц в компьютере, отличные от двойных массивов.

2. ПОСТАНОВКА ЗАДАЧИ

Мой ToDo list по данному курсовому проекту, который является структурой преокта:

[] 1. Составить структуру Матрица на массивах на трех векторах: массив значений, массив индексов столбцов, массив индексации строк.

Массив индексации строк указывает, с какого до какого индекса находятся элементы в массиве столбцов.

[] 2. Считывать значения из ввода и сохранять в матрице.

[] 3. Выводить значения матрицы в обычном виде и в виде трех массивов.

[] 4. Реализовать функцию поиска строки с наибольшим числом элементов.

[] 5. Реализовать функцию хранения массива в бинарном файле.

3. СТРУКТУРА ПРОЕКТА

Проект состоит из 4 файлов: `main.c` - главный файл, `matrix.h` - библиотека, `data1.txt` - текстовый файл с обычной матрицей, `data1.bin` - бинарный файл матрицы.

Пример работы программы ***input.c***:

```
[revammark@Raft kp7]$ date
```

```
Чт 30 апр 2020 12:13:38 MSK
```

```
[revammark@Raft kp7]$ ./main
```

Программа разреженные матрицы. Курсовой проект VII.

Выберите дальнейшее действие:

1. Считать матрицу из текстового файла
2. Считать матрицу из бинарного файла
3. Вывести матрицу
4. Вывести структуру матрицы
5. Сохранить матрицу в бинарный файл
6. Выполнить задание 7 над матрицей, т.е найти строки с наибольшим числом элементов
0. Выйти из программы

Ваш выбор: 1

Введите путь к файлу: `data1.txt`

Считал!

Выберите дальнейшее действие:

1. Считать матрицу из текстового файла
2. Считать матрицу из бинарного файла
3. Вывести матрицу
4. Вывести структуру матрицы
5. Сохранить матрицу в бинарный файл
6. Выполнить задание 7 над матрицей, т.е найти строки с наибольшим числом элементов
0. Выйти из программы

Ваш выбор: 3

4 6 11
0 2 0 0 0 0
7 0 0 9 0 0
10 16 0 0 66 69
-1 -2 -3 -4 0 0

Выберите дальнейшее действие:

1. Считать матрицу из текстового файла
2. Считать матрицу из бинарного файла
3. Вывести матрицу
4. Вывести структуру матрицы
5. Сохранить матрицу в бинарный файл
6. Выполнить задание 7 над матрицей, т.е найти строки с наибольшим числом элементов
0. Выйти из программы

Ваш выбор: 4

Rows: 4 Columes: 6 Nozeros:11

0 1 3 7 11
1 0 3 0 1 4 5 0 1 2 3
2 7 9 10 16 66 69 -1 -2 -3 -4

Выберите дальнейшее действие:

1. Считать матрицу из текстового файла
2. Считать матрицу из бинарного файла
3. Вывести матрицу
4. Вывести структуру матрицы
5. Сохранить матрицу в бинарный файл
6. Выполнить задание 7 над матрицей, т.е найти строки с наибольшим числом элементов
0. Выйти из программы

Ваш выбор: 5

Введите путь к файлу: data1.bin

Сохранил!

Выберите дальнейшее действие:

1. Считать матрицу из текстового файла
2. Считать матрицу из бинарного файла
3. Вывести матрицу
4. Вывести структуру матрицы
5. Сохранить матрицу в бинарный файл
6. Выполнить задание 7 над матрицей, т.е найти строки с наибольшим числом элементов
0. Выйти из программы

Ваш выбор: 2

Введите путь к файлу: data1.bin

Считал!

Выберите дальнейшее действие:

1. Считать матрицу из текстового файла
2. Считать матрицу из бинарного файла
3. Вывести матрицу
4. Вывести структуру матрицы
5. Сохранить матрицу в бинарный файл
6. Выполнить задание 7 над матрицей, т.е найти строки с наибольшим числом элементов
0. Выйти из программы

Ваш выбор: 6

Наибольшее число элементов в строке: 4

Строки с наибольшим числом элементов: 2 3

Выберите дальнейшее действие:

1. Считать матрицу из текстового файла
2. Считать матрицу из бинарного файла
3. Вывести матрицу
4. Вывести структуру матрицы
5. Сохранить матрицу в бинарный файл
6. Выполнить задание 7 над матрицей, т.е найти строки с наибольшим числом элементов
0. Выйти из программы

Ваш выбор: 3

4 6 11

0 2 0 0 0 0

7 0 0 9 0 0

10 16 0 0 66 69

-1 -2 -3 -4 0 0

Выберите дальнейшее действие:

1. Считать матрицу из текстового файла
2. Считать матрицу из бинарного файла
3. Вывести матрицу
4. Вывести структуру матрицы
5. Сохранить матрицу в бинарный файл
6. Выполнить задание 7 над матрицей, т.е найти строки с наибольшим числом элементов
0. Выйти из программы

Ваш выбор: 0

[revammark@Raft kp7]\$

4. ПРОГРАММНЫЙ КОД И ТЕСТОВЫЕ ДАННЫЕ

Код программы *input.c*:

```
#include <stdio.h>
#include <stdlib.h>
#include "matrix.h"

int main() {
    Matrix matrix;
    printf("Программа разреженные матрицы. Курсовой проект VII.\n");

    int ans = 0;
    char* filepath = malloc(sizeof(char)*256);
    do {
        printf("\nВыберите дальнейшее действие:\n");
        printf("1. Считать матрицу из текстового файла\n");
        printf("2. Считать матрицу из бинарного файла\n");
        printf("3. Вывести матрицу\n");
        printf("4. Вывести структуру матрицы\n");
        printf("5. Сохранить матрицу в бинарный файл\n");
        printf("6. Выполнить задание 7 над матрицей, т.е найти строки с\n");
        printf("наибольшим числом элементов\n");
        printf("0. Выйти из программы\n");
        printf("Ваш выбор: ");
        scanf("%d", &ans);

        switch (ans) {
            case 1:
                printf("Введите путь к файлу: ");
                scanf("%s", filepath);
                inputMatrix(&matrix, filepath);
                printf("Считал!");
                break;
            case 2:
                printf("Введите путь к файлу: ");
                scanf("%s", filepath);
                loadMatrix(&matrix, filepath);
                printf("Считал!");
                break;
            case 3:
                mathPrintMatrix(&matrix);
                break;
            case 4:
```

```

        printMatrix(&matrix);
        break;
    case 5:
        printf("Введите путь к файлу: ");
        scanf("%s", filepath);
        saveMatrix(&matrix, filepath);
        printf("Сохранил!");
        break;
    case 6:
        task7(&matrix);
        break;
    case 0:
        break;
    default:
        printf("Ошибочный ввод\n");
        break;
    }
} while(ans);
free(filepath);
return 0;
}

```

Код программы *studentpc.h*:

```

#ifndef __matrix_H__
#define __matrix_H__

#include <stdio.h>
#include <stdlib.h>

typedef long T;
typedef struct {
    int rows;
    int columns;
    int nozeros;
    int* index_columns;
    int* indexation_rows;
    T* data;
} Matrix;

long char_to_number(char buffer) {
    return buffer - '0';
}

long str_to_number(char* str, int size) {
    long result = 0;
    for(int i=0; i < size; i++) {
        result *= 10;
    }
}

```

```

        result += char_to_number(str[i]);
    }

    return result;
}

void inputMatrix(Matrix* matrix, char* filepath) {
    FILE* inputfile;
    inputfile = fopen(filepath, "r");

    int rows;
    int columns;
    int nozeros;
    fscanf(inputfile, "%d %d %d\n", &rows, &columns, &nozeros);
    matrix->columns = columns;
    matrix->rows = rows;
    matrix->nozeros = nozeros;
    matrix->index_columns = malloc(sizeof(int)*nozeros);
    matrix->indexation_rows = malloc(sizeof(int)*rows);
    matrix->data = malloc(sizeof(T)*nozeros);

    matrix->indexation_rows[0] = 0;
    int nums = 0;
    for(int j=0; j < rows; j++) {
        for(int i=0; i < columns; i++) {
            T num;
            fscanf(inputfile, "%ld", &num);
            if(num != 0) {
                matrix->index_columns[nums] = i;
                matrix->data[nums] = num;
                nums++;
            }
        }
        matrix->indexation_rows[j+1] = nums;
    }
    return;
}

void printMatrix(Matrix* matrix) {
    printf("Rows: %d\tColumns: %d\tNozeros:%d\n", matrix->rows, matrix->columns,
matrix->nozeros);
    for(int i=0; i <= matrix->rows; i++) {
        printf("%d ", matrix->indexation_rows[i]);
    }
    printf("\n");
    for(int i=0; i < matrix->nozeros; i++) {
        printf("%d ", matrix->index_columns[i]);
    }
    printf("\n");
    for(int i=0; i < matrix->nozeros; i++) {
        printf("%ld ", matrix->data[i]);
    }
    printf("\n");
}
}

```

```

void mathPrintMatrix(Matrix* matrix) {
    int iter = 0;
    printf("%d %d %d\n", matrix->rows, matrix->columns, matrix->nozeros);
    for(int row=0; row < matrix->rows; row++) {
        for(int colum=0; colum < matrix->columns; colum++) {
            if(matrix->indexation_rows[row] <= iter && iter < matrix-
>indexation_rows[row+1]
            && colum == matrix->index_columns[iter]) {
                printf("%ld ", matrix->data[iter]);
                iter++;
            } else {
                printf("0 ");
            }
        }
        printf("\n");
    }
    return;
}

void saveMatrix(Matrix* matrix, char* filepath) {
    FILE* outfile;
    outfile = fopen(filepath, "wb");

    fwrite(&matrix, sizeof(matrix), 1, outfile);
    return;
}

void loadMatrix(Matrix* matrix, char* filepath) {
    FILE* inputfile;
    inputfile = fopen(filepath, "rb");

    fread(&matrix, sizeof(matrix), 1, inputfile);
    return;
}

// Найти строку, содержащую наибольшее число элементов.
// Напечатать номер строки и количество элементов.
// Если таких строк несколько - обработать все
void task7(Matrix* matrix) {
    int maxelems = 0;
    int elemsInRow = 0;
    for(int i=0; i < matrix->rows; i++) {
        elemsInRow = matrix->indexation_rows[i+1] - matrix->indexation_rows[i];
        if(maxelems < elemsInRow) maxelems = elemsInRow;
    }
    printf("Наибольшее число элементов в строке: %d\n", maxelems);
    printf("Строки с наибольшим числом элементов: ");
    for(int i=0; i < matrix->rows; i++) {
        elemsInRow = matrix->indexation_rows[i+1] - matrix->indexation_rows[i];
        if(elemsInRow == maxelems) {
            printf("%d ", i);
        }
    }
    printf("\n");
    return;
}

```

```
}
```

```
#endif
```

Данные текстового файла ***data1.txt***:

```
4 6 11  
0 2 0 0 0  
7 0 0 9 0  
10 16 0 0 66 69  
-1 -2 -3 -4 0 0
```

5. ЗАКЛЮЧЕНИЕ

Я составил последовательную структуру данных для представления матриц в СП Си. Реализовал программу, сохраняющую данные в бинарный файл, и программу, выводящую данные из бинарного файла в приемлимый для пользователя вид.

Я выполнил задание, то есть реализовать функцию поиска строк с наибольшим числом элементов и выводением индекса этих строк

Я освоил работу со структурами в Си, с разреженными матрицами в СИ, работу с файлами.

6. ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Керниган Б., Ритчи Д. Язык программирования Си = The C programming language. — 2-е изд. — М.: Вильямс, 2007. — С. 304. — ISBN 0-13-110362-8.
2. Эндрю Таненбаум, *Structured Computer Organization*, [ISBN 0-13-148521-0](#)
3. В.Е. Зайцев, Конспект Лекций по курсу “Фундаментальная Информатика и Языки и методы программирования”.