



Отчёт по лабораторной работе № VII

по курсу: _____ ЯМП _____

студента группы: Чурилов С. Э. М8О-103Б-20, № по списку: 29

Адреса www, e-mail, jabber, skype churilov.ser1204@gmail.com

Работа выполнена: “ ____ ” _____ 20 ____ г.

Преподаватель: _____ каф. 806 В. К. Титов

Входной контроль знаний с оценкой _____

Отчёт сдан “ ____ ” _____ 20 ____ г., итоговая оценка _____

Подпись преподавателя _____

1 **Тема:** Разреженные матрицы

2 **Цель работы:** Составить программу для обработки прямоугольных разреженных матриц с элементами вещественного типа данных

3 **Задание (вариант № 1/7):** Цепочка ненулевых элементов в векторе А со строчным индексированием/ Найти строку, содержащую наибольшее количество ненулевых элементов, и напечатать ее номер и сумму элементов этой строки. Если таких строк несколько, обработать все.

4 **Оборудование (лабораторное):**
ЭВМ MSI GT70 0ND 447-RU, процессор Intel Celeron 2.4 GhZ, имя узла сети cameron с ОП 15 ГБ
НМД _____ ГБ. Терминал _____ адрес _____, Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Ryzen 3 3200U @ 4x 2.6GHz, ОП _____ 8192 _____ МБ, НМД _____ ГБ. Монитор: встроенный
Другие устройства _____

5 **Программное обеспечение (лабораторное):**
Операционная система семейства UNIX, наименование: Ubuntu версия 18.04.5 LTS
Интерпретатор команд: bash версия _____
Система программирования: C версия _____
Редактор текстов: Emax версия _____
Утилиты операционной системы: _____

Прикладные системы и программы: _____

Местонахождения и имена файлов программ и данных: _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства UNIX, наименование Ubuntu версия 20.04 LTS
Интерпретатор команд: bash версия _____
Система программирования: C версия _____
Редактор текстов: Emax версия _____
Утилиты операционной системы: gcc

Прикладные системы и программы: _____

Местонахождения и имена файлов программ и данных: _____

6 Идея, метод, алгоритм решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальное описание с пред- и постусловиями)

Описание алгоритма решения:

Создаем текстовый файл, в который записываем исходную разреженную матрицу. При запуске программы запрашиваем название файла, из которого затем считываем матрицу. Размещаем матрицу по заданной схеме, при помощи функции печатаем массивы A и M . При помощи функции выводим номера строк, содержащих наибольшее количество ненулевых элементов и сумму их элементов.

Сведения о написанных функциях:

Файл	Функция	Описание функции
vector.c	<code>int is_vector_empty (vector_int *v);</code>	Возвращает 1, если вектор пуст
	<code>int size (vector_int *v);</code>	Возвращает количество элементов в векторе
	<code>void vector_create_int (vector_int *v, int size);</code>	Создаёт вектор с заданным размером
	<code>void vector_push_back_int (vector_int *v, int c);</code>	Добавляет в конец вектора элемент, довыделяя память, если необходимо
	<code>void resize_int (vector_int *v);</code>	Вспомогательная функция для <code>vector_push_back_int()</code> , довыделяет память для вектора
	<code>void vector_print_int (vector_int *v);</code>	Печатает все элементы вектора через пробел
main.c	<code>int solve (int Alen, int n, int *A, int *M)</code>	Находит строки, содержащие наибольшее количество ненулевых элементов, печатает их номера и суммы их элементов
	<code>int print_M (int n, int *M)</code>	Печатает массив M
	<code>int print_A (int A_lenght, int *A)</code>	Печатает массив A

7 Сценарий выполнения работы [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты, либо соображения по тестированию].

План работы:

1. Придумать тесты;
2. Создать ядро программы
3. Создать программу;
4. Провести рефакторинг
5. Отладить основную программу.

Данные	Предполагаемый ответ
<pre> 11 6 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 3 0 0 0 0 0 0 4 0 0 0 0 0 0 5 0 0 0 0 0 0 6 1 0 1 1 1 1 0 3 0 0 3 3 4 4 0 4 4 4 5 0 5 5 1 5 </pre>	<pre> Line 8 has 5 elements, their sum = 5 Line 10 has 5 elements, their sum = 20 Line 11 has 5 elements, their sum = 21 </pre>
<pre> 5 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 </pre>	<pre> Natural elements not found </pre>
<pre> 10 10 0 0 0 2 0 0 0 0 0 0 7 0 0 0 0 0 4 0 3 0 5 0 0 6 0 0 0 0 0 0 5 0 5 0 2 1 5 0 0 0 0 0 0 0 0 0 3 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 1 0 0 0 </pre>	<pre> Line 2 has 3 elements, their sum = 14 Line 7 has 3 elements, their sum = 8 </pre>
<pre> 8 10 0 0 0 0 0 0 0 6 0 0 0 1 0 0 0 0 5 0 0 0 3 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 7 0 0 0 0 4 0 0 0 0 0 0 0 0 0 3 </pre>	<pre> Line 2 has 2 elements, their sum = 6 Line 3 has 2 elements, their sum = 10 Line 7 has 2 elements, their sum = 11 </pre>
<pre> 4 3 1 2 0 0 3 4 5 0 6 0 0 0 </pre>	<pre> Line 1 has 2 elements, their sum = 3 Line 2 has 2 elements, their sum = 7 Line 4 has 2 elements, their sum = 11 </pre>
<pre> 5 5 0 0 0 0 0 3 0 6 0 1 3 1 0 0 0 0 0 0 0 0 0 0 0 0 0 </pre>	<pre> Line 2 has 3 elements, their sum = 10 </pre>

Допущен к выполнению работы. Подпись преподавателя _____

8 Распечатка протокола (подклеить листинг окончательного варианта программы с текстовыми примерами, подписанный преподавателем)

```
leninware@leninware-VirtualBox:~/LRVII$ date
Чт 20 май 2021 10:43:44 MSK
leninware@leninware-VirtualBox:~/LRVII$ pwd
/home/leninware/LRVII
leninware@leninware-VirtualBox:~/LRVII$ ls
about input input2 input3 input4 input5 input6 main.c vector.c vector.h P6789-202021.pdf
leninware@leninware-VirtualBox:~/LRVII$ cat about
*****
ФИО:Чурилов Сергей Эдуардович
Группа:M8O-103Б-20
E-mail: churilov.ser1204@gmail.com
ЛР: VII
Номер по списку: 29
*****
leninware@leninware-VirtualBox:~/LRVII$ cat vector.h
#ifndef VECTOR_H
#define VECTOR_H

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

typedef struct vector_int {
    int number_of_elements;
    int capacity;
    int *elem;
} vector_int;

int is_vector_empty(vector_int *v);
int size(vector_int *v);
void vector_create_int(vector_int *v, int size);
void vector_push_back_int(vector_int *v, int c);
void resize_int(vector_int *v);
void vector_print_int(vector_int *v);

#endif
leninware@leninware-VirtualBox:~/LRVII$ cat vector.c
#include "vector.h"

int is_vector_empty(vector_int *v)
{
    if (v->number_of_elements == 0) {
        return 1;
    }
    return 0;
}

int size(vector_int *v)
{
    return v->capacity;
}

void vector_create_int(vector_int *v, int size)
{
    v->capacity = size;
    v->number_of_elements = 0;
    v->elem = (int*)malloc(sizeof(int) * v->capacity);
}

void vector_push_back_int(vector_int *v, int c)
{

```

```

    if (v->number_of_elements == v->capacity) {
        resize_int(v);
    }
    v->elem[v->number_of_elements - 1] = c;
    v->number_of_elements++;
}

void resize_int(vector_int *v)
{
    v->capacity++;
    v->elem = (int*)realloc(v->elem, sizeof(int) * (v->capacity + 1));
}

```

```

void vector_print_int(vector_int *v)
{
    for (int i = -1; i < v->number_of_elements - 1; ++i) {
        printf("%d ", v->elem[i]);
    }
}

leninware@leninware-VirtualBox:~/LRVII$ cat main.c
#include <stdio.h>
#include <string.h>
#include "vector.h"

```

```

const int INF = 1e9 + 7;

```

```

int solve(int A_lenght, int n, int *A, int *M)
{
    int max_wide = 0, widest_line, tmp = -1, curSUM = 0;
    vector_int widers;
    vector_create_int(&widers, 2);
    for (int i = n - 1; i >= 0; --i) {
        if (M[i] - M[i - 1] >= max_wide && M[i - 1] != 0) {
            max_wide = M[i] - M[i - 1];
        }
    }
    int last = A_lenght;
    for (int i = 0; i < 2; ++i) {
        if (last - M[n - i] >= max_wide && M[n - i] != 0) {
            max_wide = A_lenght - M[n - i];
            tmp = n;
            break;
        }
        last = M[n - i];
    }
    for (int i = 0; i <= n; ++i) {
        if (M[i] == 0) {
            continue;
        }
        if (M[i + 1] == 0) {
            for (int j = i + 1; j < n; ++j) {
                if (M[j] - M[i] == max_wide) {
                    vector_push_back_int(&widers, i + 1);
                }
                if (M[j] != 0) break;
            }
        }
        if (M[i + 1] - M[i] == max_wide) {
            vector_push_back_int(&widers, i + 1);
        }
    }
    if (tmp != -1) {
        vector_push_back_int(&widers, tmp);
    }
}

```

```

    }
    max_wide /= 3;
    int SUM[n], sum = 0, wide = 0;
    for (int i = 1; i < A_lenght; ++i) {
        int a1 = i;
        ++i;
        int a2 = i;
        ++i;
        int a3 = i;
        sum += A[a2];
        ++wide;
        if (A[a3] == 0 && wide == max_wide) {
            SUM[curSUM] = sum;
            ++curSUM;
            sum = 0;
        }
        if (A[a3] == 0) {
            wide = 0;
            sum = 0;
        }
    }
    for (int i = -1; i < (&widers)->number_of_elements - 1; ++i) {
        if (max_wide == 0) {
            printf("Natural elements not found\n");
        } else {
            printf("Line %d has %d elements, their sum = %d\n", (&widers)->elem[i], max_wide, SUM[i + 1]);
        }
    }
    return max_wide;
}

```

```

int print_M(int n, int *M)
{
    printf("M: ");
    for (int i = 0; i < n; ++i) {
        printf("%d ", M[i]);
    }
    printf("\n");
}

```

```

int print_A(int A_lenght, int *A)
{
    printf("A: ");
    for (int i = 0; i < A_lenght; ++i) {
        printf("%d ", A[i]);
    }
    printf("\n");
}

```

```

int main(int argc, char *argv[])
{
    FILE *file;
    char name[50];
    int n, m, A_lenght = 1;
    if (argc > 1) {
        file = fopen(argv[1], "r");
    } else {
        printf("Enter the file's name: ");
        scanf("%s", name);
        file = fopen(name, "r");
    }
    fscanf(file, "%d%d", &n, &m);
    int mtx[n][m];
}

```

```

printf("n = %d, m = %d\n", n, m);
printf("Your matrix:\n");
for (int i = 0; i < n; ++i) {
    for (int j = 0; j < m; ++j) {
        fscanf(file, "%d", &mtx[i][j]);
        printf("%d ", mtx[i][j]);
        if (mtx[i][j] != 0) {
            A_lenght += 3;
        }
    }
    printf("\n");
} // end of input

// create A[] and M[]
int M[n], A[A_lenght];
for (int i = 0; i < n; ++i) {
    M[i] = 0;
}
for (int i = 0; i < A_lenght; ++i) {
    A[i] = -INF;
}

// fill A[] and M[]
int count_in_line = 0, max_in_line = 0, a1 = 1, a2 = 2, a3 = 3, a3_changed = 0, M_changed = 0;
for (int i = 0; i < n; ++i) {
    for (int j = 0; j < m; ++j) {
        if (mtx[i][j] != 0) {
            if (a3_changed) {
                A[a3] = a1; a3 += 3;
            }
            ++count_in_line;
            if (!M_changed) {
                A[a3 - 3] = 0;
                M[i] = a1;
                M_changed = 1;
            }
            A[a1] = j; a1 += 3;
            A[a2] = mtx[i][j]; a2 += 3;
            A[a3] = 0;
            a3_changed = 1;
        }
    }
    if (max_in_line < count_in_line) {
        max_in_line = count_in_line;
    }
    if (count_in_line == 0) {
        M[i] = 0;
    }
    count_in_line = 0;
    M_changed = 0;
}
A[0] = 0;

// print M and A
print_M(n, M);
print_A(A_lenght, A);

// function via A[] and M[]
solve(A_lenght, n, A, M);
return 0;
}

```

```
leninware@leninware-VirtualBox:~/LRVII$ cat input
11 6
000000
000000
020000
003000
000400
000050
000006
101111
030033
440444
505515
leninware@leninware-VirtualBox:~/LRVII$ cat input2
5 3
000
000
000
000
000
leninware@leninware-VirtualBox:~/LRVII$ cat input3
10 10
0002000000
7000004030
5006000000
5000000000
0000000000
0000000050
2150000000
0030040000
0000000000
0060001000
leninware@leninware-VirtualBox:~/LRVII$ cat input4
8 10
0000000600
0100005000
3000007000
0000000000
0400000000
0000000000
0000700004
0000000003
leninware@leninware-VirtualBox:~/LRVII$ cat input5
4 3
120
034
506
000
leninware@leninware-VirtualBox:~/LRVII$ cat input6
5 5
00000
30601
31000
00000
00000
leninware@leninware-VirtualBox:~/LRVII$ g++ main.cpp -o main
leninware@leninware-VirtualBox:~/LRVII$ ./main < input
n = 11, m = 6
Your matrix:
000000
000000
020000
003000
```



```
000400
000050
000006
101111
030033
440444
505515
M: 00147101316314055
A: 012023034045056001192122312541285101334433753004431446344944
525400558256135644167550
Line 8 has 5 elements, their sum = 5
Line 10 has 5 elements, their sum = 20
Line 11 has 5 elements, their sum = 21
leninware@leninware-VirtualBox:~/LRVII$ ./main < input2
n = 5, m = 3
Your matrix:
000
000
000
000
000
M: 00000
A: 0
Natural elements not found
leninware@leninware-VirtualBox:~/LRVII$ ./main < input3
n = 10, m = 10
Your matrix:
0002000000
7000004030
5006000000
5000000000
0000000000
0000000050
2150000000
0030040000
0000000000
M: 1413190222534040
A: 0320077641083005163600508500228113125023375402643610
Line 2 has 3 elements, their sum = 14
Line 7 has 3 elements, their sum = 8
leninware@leninware-VirtualBox:~/LRVII$ ./main < input4
n = 8, m = 10
Your matrix:
0000000600
0100005000
3000007000
0000000000
0400000000
0000000000
0000700004
0000000003
M: 141001601925
A: 076011765003136701404722940930
Line 2 has 2 elements, their sum = 6
Line 3 has 2 elements, their sum = 10
Line 7 has 2 elements, their sum = 11
leninware@leninware-VirtualBox:~/LRVII$ ./main < input5
n = 4, m = 3
Your matrix:
120
034
506
000
```

M: 1 7 13 0

A: 0 0 1 4 1 2 0 1 3 10 2 4 0 0 5 16 2 6 0

Line 1 has 2 elements, their sum = 3

Line 2 has 2 elements, their sum = 7

Line 4 has 2 elements, their sum = 11

leninware@leninware-VirtualBox:~/LRVII\$./main < input6

n = 5, m = 5

Your matrix:

0 0 0 0 0

3 0 6 0 1

3 1 0 0 0

0 0 0 0 0

0 0 0 0 0

M: 0 1 10 0 0

A: 0 0 3 4 2 6 7 4 1 0 0 3 13 1 1 0

Line 2 has 3 elements, their sum = 10

leninware@leninware-VirtualBox:~/LRVII\$

- 9 **Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	дом	20.05.2021	10:00	неправильная передача двумерного массива по ссылке	добавить еще один знак *	

- 10 Замечание автора по существу работы _____

- 11 Выводы Благодаря выполнению данной лабораторной работы я научился работать с разряженными матрицами во внутреннем представлении.

Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом _____

Подпись студента _____

