

Отчет по Курсовой работе № 7

по курсу: 1 фундаментальная информатика

студент группы : М8О-105Б-21 Титеев Рамиль Маратович , № по списку: 23

Адреса www, e-mail, jabber, skype: derol.gym@gmail.com

Работа выполнена: "17 апреля 2022г"

Преподаватель: каф. 806 В.К.Титов

Входной контроль знаний с оценкой: _____

Отчет сдан " __ " _____ 20__ г., итоговая оценка _____

Подпись преподавателя _____

1. Тема: Разреженные матрицы.

2. Цель работы: Составить программу на языке Си с процедурами и/или функциями для обработки прямоугольных разреженных матриц с элементами целого, вещественного, или комплексного типов.

3. Задание: Вариант схемы размещения матрицы: 3. Три вектора;

Вариант преобразований: Определить максимальный по модулю элемент матрицы и разделить на него все элементы строки, в которой он находится. Если таких элементов несколько, обработать каждую строку, содержащую такой элемент.

4. Оборудование(лабораторное):

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ ГБ

НМД _____ ГБ. Терминал _____ адрес _____. Принтер _____

Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Ryzen 4600 @ 6x 3.0 GHz , ОП 16384 МБ, НМД _____ ГБ. Монитор Встроенный

Другие устройства _____

5. Программное обеспечение(лабораторное):

Операционная система семейства UNIX, наименование _____ версия _____

Интерпретатор команд: _____ версия _____

Система программирования: _____ версия _____

Редактор текстов: _____ версия _____

Утилиты операционной системы: _____

Прикладные системы и программы: _____

Местонахождение и имена файлов и программ данных: _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства UNIX, наименование Ubuntu версия 20.04

Интерпретатор команд: bash версия _____

Система программирования: C версия _____

Редактор текстов: Emacs версия _____

Утилиты операционной системы: _____

Прикладные системы и программы: _____

Местонахождение и имена файлов и программ данных: /usr/bin , а также /bin

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

На вход программе подается файл содержащий необходимые данные. В них входят размер прямоугольной программы, количество ненулевых элементов, и сами данные матрицы представленные в трех векторах. Параллельно вводу матрицы происходит поиск модуля максимального значения

После того как произойдет считывание всех данных, программы начинается работу. Сначала выводится матрица в виде трех векторов, какими она была введена, затем она преобразуется в прямоугольную матрицу и тоже выводится в терминал.

Затем начинается преобразование. Проходимся по строкам прямоугольной матрицы, и если встретилось число, по модулю равное максимальному, то делим всю строку на это значение.

Затем выводим преобразованную прямоугольную матрицу, конвертируем её в сокращенный вид и выводим его тоже.

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

curs_7.cpp:

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

// convert full to short
void ftosh(float ** matrix, int n, int m, int *cip, int *pi, float *ye){
    int k = 0;
    for (int i = 0; i<n; i++){
        int was_no_zero = 0;
        for (int j = 0; j<m; j++){
            if (matrix[i][j]!=0){
                ye[k] = matrix[i][j];
                pi[k] = j;
                if (!was_no_zero){
                    was_no_zero = 1;
                    cip[i] = k;
                }
                k++;
            }
        }
        if (!was_no_zero){
            cip[i] = -1;
        }
    }
}

// convert short to full
void shtof (float ** matrix, int n, int *cip, int *pi, float *ye , int k){
    int s;
    int start, last_i;
    for (s = 0; cip[s] == -1 && s<n; ++s);

    for (int i = s; i<n; ++i){
        start = cip[i];
        if (start == -1){
            continue;
        }
        int j;
        for (j = i+1; cip[j] == -1 && j<n-1; j++);
        last_i = i;
```

```

        for (int l = start; l < cip[j]; l++){
            matrix[i][pi[l]] = ye[l];
        }
    }
    for (int l = cip[last_i]; l < k; l++){
        matrix[last_i][pi[l]] = ye[l];
    }
}

// convert full matrix
void print_f_matrix(float ** matrix, int n, int m){
    printf("Full matrix:\n");
    for (int i = 0; i<n; i++){
        for (int j = 0; j<m; j++){
            printf("%2.2f ", matrix[i][j]);
        }
        printf("\n");
    }
}

// convert short matrix
void print_sh_matrix(int *cip, int *pi, float *ye, int k, int n){
    printf("Short matrix:\n");
    printf("CIP: ");
    for (int i = 0; i<n; i++){
        printf("%d ", cip[i]);
    }
    printf("\n");
    printf("PI: ");
    for (int i = 0; i<k+1; i++){
        printf("%d ", pi[i]);
    }
    printf("\n");
    printf("YE: ");
    for (int i = 0; i<k; i++){
        printf("%2.2f ", ye[i]);
    }
    printf("\n");
}

// transform matrix matrix
void transform_matrix(float ** matrix, int n, int m, int max_v){
    for (int i = 0; i< n; ++i){
        for (int j = 0; j< m; ++j){
            if (matrix[i][j] == max_v || matrix[i][j] == -max_v){
                for (int c = 0; c< m; ++c){
                    matrix[i][c] /= max_v;
                }
                break;
            }
        }
    }
}

```

```

int main(){
    float **matrix, *ye;
    int *cip, *pi;
    int n, m, k;
    printf("Input n = \n"); scanf("%d",&n);
    printf("Input m =\n"); scanf("%d",&m);
    printf("Input k =\n"); scanf("%d",&k);
    matrix = new float*[n];
    pi = new int[k+1];
    ye = new float[k];
    cip = new int[n];
    int max_v = 0;

    for (int i = 0; i<n; i++){
        matrix[i] = new float[m];
    }

    printf("Input short matrix:\n");
    printf("CIP: ");
    for (int i = 0; i<n; i++){
        scanf("%d", &cip[i]);
    }
    printf("\n");
    printf("PI: ");
    for (int i = 0; i<k+1; i++){
        scanf("%d", &pi[i]);
    }
    printf("\n");
    printf("YE: ");
    for (int i = 0; i<k; i++){
        scanf("%f", &ye[i]);
        if (ye[i] > max_v || -ye[i] > max_v){
            max_v = ye[i];
        }
    }
}

if (max_v < 0) max_v = -max_v;

printf("\n");
printf("\n");
print_sh_matrix(cip, pi, ye, k, n);
shtof (matrix, n, cip, pi, ye, k);
print_f_matrix(matrix, n, m);
transform_matrix(matrix, n, m, max_v);

printf("\n");
printf("After transform:\n");
printf("\n");
print_f_matrix(matrix, n, m);
ftosh(matrix, n, m, cip, pi, ye);
print_sh_matrix(cip, pi, ye, k, n);

```

```

return 0;

}

```

Пункты 1-7 отчёта составляются **строго до** начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

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

```

constantfear@constantfear:~/projects/laboratory/2_semester/curs_7$ cat header.txt
*****
*                               *
*           Курсовая работа №7           *
*           Разрезанные матрицы.           *
*           Выполнил студент гр. М80-105-Б           *
*           Титеев Рамиль Маратович           *
*                               *
*****
constantfear@constantfear:~/projects/laboratory/2_semester/curs_7$ cat in1
4 5 4
0 -1 1 2
3 1 2 4 0
1 4 2 7
constantfear@constantfear:~/projects/laboratory/2_semester/curs_7$ cat in2
11 9 18
0 2 -1 4 6 9 11 14 -1 15 17
4 6 1 7 2 6 0 3 6 5 8 0 1 6 7 3 8 5 0
-9 1 1 2 9 8 4 1 8 -9 3 2 2 1 1 4 9 1
constantfear@constantfear:~/projects/laboratory/2_semester/curs_7$ cat in3
6 5 5
0 -1 1 3 4 -1
4 0 1 1 3 0
1 2 4 3 -4
constantfear@constantfear:~/projects/laboratory/2_semester/curs_7$ g++ curs_7.cpp
constantfear@constantfear:~/projects/laboratory/2_semester/curs_7$ ./a.out < in1
Input n =
Input m =
Input k =
Input short matrix:
CIP:
PI:
YE:

Short matrix:
CIP: 0 -1 1 2
PI: 3 1 2 4 0
YE: 1.00 4.00 2.00 7.00
Full matrix:
0.00 0.00 0.00 1.00 0.00
0.00 0.00 0.00 0.00 0.00
0.00 4.00 0.00 0.00 0.00
0.00 0.00 2.00 0.00 7.00

After transform:

Full matrix:
0.00 0.00 0.00 1.00 0.00
0.00 0.00 0.00 0.00 0.00
0.00 4.00 0.00 0.00 0.00
0.00 0.00 0.29 0.00 1.00
Short matrix:
CIP: 0 -1 1 2

```

```

PI: 3 1 2 4 0
YE: 1.00 4.00 0.29 1.00
constantfear@constantfear:~/projects/laboratory/2_semester/curs_7$ ./a.out < in2
Input n =
Input m =
Input k =
Input short matrix:
CIP:
PI:
YE:

Short matrix:
CIP: 0 2 -1 4 6 9 11 14 -1 15 17
PI: 4 6 1 7 2 6 0 3 6 5 8 0 1 6 7 3 8 5 0
YE: -9.00 1.00 1.00 2.00 9.00 8.00 4.00 1.00 8.00 -9.00 3.00 2.00 2.00 1.00 1.00 4.00 9.00 1.00
Full matrix:
0.00 0.00 0.00 0.00 -9.00 0.00 1.00 0.00 0.00
0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 2.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 9.00 0.00 0.00 0.00 8.00 0.00 0.00
4.00 0.00 0.00 1.00 0.00 0.00 8.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 -9.00 0.00 0.00 3.00
2.00 2.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 4.00 0.00 0.00 0.00 0.00 9.00
0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00

After transform:

Full matrix:
0.00 0.00 0.00 0.00 -1.00 0.00 0.11 0.00 0.00
0.00 1.00 0.00 0.00 0.00 0.00 0.00 2.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 1.00 0.00 0.00 0.00 0.89 0.00 0.00
4.00 0.00 0.00 1.00 0.00 0.00 8.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 -1.00 0.00 0.00 0.33
2.00 2.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.44 0.00 0.00 0.00 0.00 1.00
0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00
Short matrix:
CIP: 0 2 -1 4 6 9 11 14 -1 15 17
PI: 4 6 1 7 2 6 0 3 6 5 8 0 1 6 7 3 8 5 0
YE: -1.00 0.11 1.00 2.00 1.00 0.89 4.00 1.00 8.00 -1.00 0.33 2.00 2.00 1.00 1.00 0.44 1.00 1.00
constantfear@constantfear:~/projects/laboratory/2_semester/curs_7$ ./a.out < in3
Input n =
Input m =
Input k =
Input short matrix:
CIP:
PI:
YE:

Short matrix:
CIP: 0 -1 1 3 4 -1
PI: 4 0 1 1 3 0
YE: 1.00 2.00 4.00 3.00 -4.00
Full matrix:
0.00 0.00 0.00 0.00 1.00
0.00 0.00 0.00 0.00 0.00
2.00 4.00 0.00 0.00 0.00

```

```

0.00 3.00 0.00 0.00 0.00
0.00 0.00 0.00 -4.00 0.00
0.00 0.00 0.00 0.00 0.00

```

After transform:

Full matrix:

```

0.00 0.00 0.00 0.00 1.00
0.00 0.00 0.00 0.00 0.00
0.50 1.00 0.00 0.00 0.00
0.00 3.00 0.00 0.00 0.00
0.00 0.00 0.00 -1.00 0.00
0.00 0.00 0.00 0.00 0.00

```

Short matrix:

CIP: 0 -1 1 3 4 -1

PI: 4 0 1 1 3 0

YE: 1.00 0.50 1.00 3.00 -1.00

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

10. Замечание автора по существу работы _____

11. Выводы _____ Я научился работать с разреженными матрицами.

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

Подпись студента _____