

Отчет по лабораторной работе № VII по курсу «Практикум на ЭВМ»

Студент группы М8О-105Б-21 Минеева Светлана Алексеевна, № по списку 14

Контакты e-mail: svetlana.mineewa2003@yandex.ru

Работа выполнена: «19» апреля 2022 г.

Преподаватель: Вячеслав Константинович Титов каф. 805

Отчет сдан «19» апреля 2022 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 вещественного типа, которая:

1. Вводит матрицы различного размера, представленные во входном текстовом файле в обычном формате (по строкам), с одновременным размещением ненулевых элементов в разреженной матрице в соответствии с заданной схемой;
2. Печатает введенные матрицы во внутреннем представлении согласно заданной схеме размещения и в обычном (естественном) виде;
3. Выполняет необходимые преобразования разреженных матриц (или вычисления над ними) путем обращения к соответствующим процедурам и/или функциям;
4. Печатает результат преобразования (вычисления) согласно заданной схеме размещения и в обычном виде.

3. Задание (вариант № 2,5):

Вариант схемы размещения матрицы: 2. Один вектор:

Ненулевому элементу соответствуют две ячейки: первая содержит номер столбца, вторая содержит значение элемента. Нуль в первой ячейке означает конец строки, а вторая ячейка содержит в этом случае номер следующей хранимой строки. Нули в обеих ячейках являются признаком конца перечня ненулевых элементов разреженной матрицы.

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

...

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

Вариант преобразований: 5. Умножить вектор-строку на разреженную матрицу и вычислить количество ненулевых элементов результата.

4. Оборудование (лабораторное):

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор 2,9 GHz 2-ядерный процессор Intel Core i5 с ОП 8 Гб, НМД 500 Гб. Монитор 13,3-дюймовый (2560 x 1600).

Другие устройства _____

5. Программное обеспечение (лабораторное):

Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____

Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства UNIX, наименование Terminal версия 2.10

интерпретатор команд bash версия 3.2.

Система программирования _____ версия _____

Редактор текстов emacs версия 27.2-2

Утилиты операционной системы cat, ls, g++ и другие

Прикладные системы и программы _____

Местонахождение и имена файлов программ и данных на домашнем компьютере _____

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

Алгоритм работы:

На вход программе подаётся размер прямоугольной разреженной матрицы, элементы вектор-строки и один вектор, дающий нам информацию о разреженной матрице.

После ввода данных на экран выводится вектор-строка, разреженная матрица в виде вектора. После, беря данный из вектора, составляется прямоугольная расширенная матрица и выводится на экран.

Затем начинается преобразование разреженной матрицы. Умножаем вектор-строку на разреженную матрицу стандартным образом: умножаем последовательно элементы векторной строки на элементы столбца разреженной матрицы, складывая получающиеся произведения. Повторяем это действие для каждого столбца разреженной матрицы и составляем новую матрицу – результирующую матрицу, одновременно подсчитываем количество получающихся ненулевых элементов.

Проходим последовательно по результирующей матрице, составляя из нее один вектор, дающий нам о ней полную информацию (вектор составляется по правилам введенного вектора для разреженной матрицы).

Выводим этот вектор, описывающий результирующую матрицу. После выводим её стандартный прямоугольный вид и количество ненулевых элементов в ней.

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

```
#include <stdio.h>
void umn_vstr_mtr(float*, float**, int, int);
void prn_mtr(float**, int, int);
void prn_vstr(float*, int);
void prn_vec(float*);
int main(){
    float **mtr, *vstr, *vec, y=1; int m, x=1, n, i=1, k=-1;
    printf("Input m="); scanf("%d",&m);
    printf("Input n="); scanf("%d",&n);
    vstr = new float[m];
    vec = new float[1000];
    printf("Enter the vector string elements: ");
    for(i = 0; i < m; i++) scanf("%f",&vstr[i]);
    mtr = new float*[m];
    printf("Enter the matrix: ");
    for(i = 0; i < m; i++) mtr[i] = new float[n];
    while(x!=0 || y!=0){
        k+=1; scanf("%d",&x); vec[k]=x; k+=1; scanf("%f",&y); vec[k]=y;
        if (x==0 && y==0) break;
        else if (x==0 && y!=0) i=y-1;
        else mtr[i][x-1] = y;}
    prn_vstr(vstr, m);
    prn_vec(vec);
    prn_mtr(mtr, n, m);
    umn_vstr_mtr(vstr, mtr, n, m);}

void prn_vec(float *vec){
```

```

int i=0;
printf("\nShort array:\n");
while((vec[i]+vec[i+1])!=0){
    printf(" %3.2f",vec[i]);i++;}
    printf(" %3.2f",vec[i]); printf("\n");}

void prn_vstr(float *vstr, int m){
    int i; printf("\nVector string:\n");
    for(i=0; i<m; i++){
        printf(" %3.2f", vstr[i]);}
        printf("\n");}

void prn_mtr(float **mtr, int n, int m){
    int x,y; printf("\nFull matrix:");
    for(x=0; x<m; x++){
        printf("\n");
        for(y=0; y<n; y++) printf(" %3.2f", mtr[x][y]);}}

void umn_vstr_mtr(float *vstr, float **mtr, int n, int m){
    float *ummtr; int i, j, x, k=0;
    ummtr = new float[n];
    for(j=0;j<n;j++){
        ummtr[j] = 0;
        for(i=0;i<m;i++){
            ummtr[j] += vstr[i]*mtr[i][j];}}
    printf("\n\nShort resulting array:\n");
    for(x=0; x<m; x++){
        if(ummtr[x]!=0) {printf("%3d%3d",k,x+1); printf(" %3.2f",k+1,ummtr[x]);}}
    printf("%3d%3d",k,k);
    printf("\n\nFull resulting matrix:");
    for(x = 0; x < n; x++){
        printf("\n");
        printf(" %3.2f ", ummtr[x]);
        if(ummtr[x]!=0) k+=1;}
    printf("\n\nNumber of non-zero elements: "); printf("%d",k); printf("\n");}

```

Тесты:

- 1) Ввод данных из файла in7: 5 3 1.5 10.4 0 1 3 0 2 1 1 0 3 2 2 0 4 1 1.1 0 5 3 10.9 0 0

Вывод: Vector string: 1.5 10.4 0 1 3

Short array: 0 2 1 1 0 3 2 2 0 4 1 1.1 0 5 3 10.9 0 0

Full matrix: 0 0 0

1 0 0

0 2 0

1.1 0 0

0 0. 10.9

Short resulting array: 0 1 1 11.5 0 3 1 32.7 0 0

Full resulting matrix: 11.5

0

32.7

Number of non-zero elements: 2

- 2) Ввод данных из файла in71: 5 8 23.1 1 5 4 0.5 0 1 2 11.5 0 2 5 1.2 0 4 3 3 5 4 0 0

Вывод: Vector string: 23.1 1 5 4 0.5

Short array: 0 1 2 11.5 0 2 5 1.2 0 4 3 3 5 4 0 0

Full matrix: 0 11.5 0 0 0 0 0 0 0 0

0 0 0 0 0 1.2 0 0 0 0

0 0 0 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

Short resulting array: 0 2 1 265.65 0 3 1 12 0 5 1 17.2 0 0

Full resulting matrix: 0

265.65

12

0

17.2

0

0

0

Number of non-zero elements: 3

3) Ввод данных: 2 4 1 2.2 0 1 2 1.4 0 0

Вывод: Vector string: 1 2.2

Short array: 0 1 2 1.4 0 0

Full matrix: 0 1.4 0 0

0 0 0 0

Short resulting array: 0 2 1 1.4 0 0

Full resulting matrix: 0

1.4

0

0

Number of non-zero elements: 1

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

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

Last login: Tue Apr 19 11:20:47 on ttys000

The default interactive shell is now zsh.

To update your account to use zsh, please run `chsh -s /bin/zsh`.

For more details, please visit <https://support.apple.com/kb/HT208050>.

MacBook-Pro-MacBook:~ macbookpro\$ cat zag.txt

```
*****
*   Минеева Светлана Алексеевна   *
*           M80-105Б-21             *
*   Лабораторная работа №VII      *
*****
```

MacBook-Pro-MacBook:~ macbookpro\$ ls |tail -6

```
PycharmProjects
in7
in71
lab7.cpp
pslog_20200707_123036.log
zag.txt
```

MacBook-Pro-MacBook:~ macbookpro\$ cat in7

```
5 3 1.5 10.4 0 1 3 0 2 1 1 0 3 2 2 0 4 1 1.1 0 5 3 10.9 0 0
```

MacBook-Pro-MacBook:~ macbookpro\$ cat in71

```
5 8 23.1 1 5 4 0.5 0 1 2 11.5 0 2 5 1.2 0 4 3 3 5 4 0 0
```

MacBook-Pro-MacBook:~ macbookpro\$ cat lab7.cpp

```
#include <stdio.h>
void umn_vstr_mtr(float*, float**, int, int);
void prn_mtr(float**, int, int);
void prn_vstr(float*, int);
void prn_vec(float*);
int main(){
    float **mtr, *vstr, *vec, y=1; int m, x=1, n, i=1, k=-1;
    printf("Input m="); scanf("%d",&m);
    printf("Input n="); scanf("%d",&n);
    vstr = new float[m];
    vec = new float[1000];
    printf("Enter the vector string elements: ");
    for(i = 0; i < m; i++) scanf("%f",&vstr[i]);
    mtr = new float*[m];
    printf("Enter the matrix: ");
    for(i = 0; i < m; i++) mtr[i] = new float[n];
    while(x!=0 || y!=0){
        k+=1; scanf("%d",&x); vec[k]=x; k+=1; scanf("%f",&y); vec[k]=y;
        if (x==0 && y==0) break;
        else if (x==0 && y!=0) i=y-1;
        else mtr[i][x-1] = y;}
    prn_vstr(vstr, m);
    prn_vec(vec);
    prn_mtr(mtr, n, m);
    umn_vstr_mtr(vstr, mtr, n, m);}

void prn_vec(float *vec){
    int i=0;
    printf("\nShort array:\n");
    while((vec[i]+vec[i+1])!=0){
        printf("  %3.2f",vec[i]);i++;}
    printf("  %3.2f",vec[i]); printf(" %3.2f",vec[i+1]); printf("\n");}

void prn_vstr(float *vstr, int m){
    int i; printf("\nVector string:\n");
    for(i=0; i<m; i++){
        printf("  %3.2f", vstr[i]);}
    printf("\n");}

void prn_mtr(float **mtr, int n, int m){
    int x,y; printf("\nFull matrix:");
    for(x=0;x<m; x++){
```

```

        printf("\n");
        for(y=0; y<n; y++) printf("  %3.2f", mtr[x][y]);}}

void umn_vstr_mtr(float *vstr, float **mtr, int n, int m){
    float *ummtr; int i, j, x, k=0;
    ummtr = new float[n];
    for(j=0; j<n; j++){
        ummtr[j] = 0;
        for(i=0; i<m; i++){
            ummtr[j] += vstr[i]*mtr[i][j];}}
    printf("\n\nShort resulting array:\n");
    for(x=0; x<m; x++){
        if(ummtr[x]!=0) {printf("%3d%3d",k,x+1); printf("%3d  %3.2f",k+1,ummtr[x]);}}
    printf("%3d%3d",k,k);
    printf("\n\nFull resulting matrix:");
    for(x = 0; x < n; x++){
        printf("\n");
        printf("  %3.2f ", ummtr[x]);
        if(ummtr[x]!=0) k+=1;}
    printf("\n\nNumber of non-zero elements: "); printf("%d",k); printf("\n");}

```

MacBook-Pro-MacBook:~ macbookpro\$ g++ -Wall -o lab7 lab7.cpp

MacBook-Pro-MacBook:~ macbookpro\$ ls |tail -7

PycharmProjects

in7

in71

lab7

lab7.cpp

pslog_20200707_123036.log

zag.txt

MacBook-Pro-MacBook:~ macbookpro\$./lab7 < in7

Input m=Input n=Enter the vector string elements: Enter the matrix:

Vector string:

1.50 10.40 0.00 1.00 3.00

Short array:

0.00 2.00 1.00 1.00 0.00 3.00 2.00 2.00 0.00 4.00 1.00 1.10 0.00 5.00
3.00 10.90 0.00 0.00

Full matrix:

0.00 0.00 0.00
1.00 0.00 0.00
0.00 2.00 0.00
1.10 0.00 0.00
0.00 0.00 10.90

Short resulting array:

0 1 1 11.50 0 3 1 32.70 0 0

Full resulting matrix:

11.50
0.00
32.70

Number of non-zero elements: 2

MacBook-Pro-MacBook:~ macbookpro\$./lab7 < in71

Input m=Input n=Enter the vector string elements: Enter the matrix:

Vector string:

23.10 1.00 5.00 4.00 0.50

Short array:

0.00 1.00 2.00 11.50 0.00 2.00 5.00 1.20 0.00 4.00 3.00 3.00 5.00 4.00
0.00 0.00

Full matrix:

0.00 11.50 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 1.20 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 3.00 0.00 4.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

Short resulting array:

```
0 2 1 265.65 0 3 1 12.00 0 5 1 17.20 0 0
```

Full resulting matrix:

```
0.00
265.65
12.00
0.00
17.20
0.00
0.00
0.00
```

Number of non-zero elements: 3

MacBook-Pro-MacBook:~ macbookpro\$./lab7

Input m=2

Input n=4

Enter the vector string elements: 1 2.2

Enter the matrix: 0 1 2 1.4 0 0

Vector string:

```
1.00 2.20
```

Short array:

```
0.00 1.00 2.00 1.40 0.00 0.00
```

Full matrix:

```
0.00 1.40 0.00 0.00
0.00 0.00 0.00 0.00
```

Short resulting array:

```
0 2 1 1.40 0 0
```

Full resulting matrix:

```
0.00
1.40
0.00
0.00
```

Number of non-zero elements: 1

MacBook-Pro-MacBook:~ macbookpro\$

9. Дневник отладки должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) 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.	19.04.22	11:00	Неверно составлена матрица	Ошибка исправлена; матрица составляется верно	При заполнении матрицы неверно были указаны индексы, так как индексы начинаются с нуля нужно вычитать из x единицу <code>mtr[i][x-1]</code>

10. Замечания автора по существу работы

Эта лабораторная работа очень полезна, она отлично развивает мышление и учит программированию на языке Си.

11. Выводы

Я составила программу на языке Си с процедурами для обработки прямоугольных разреженных матриц с элементами вещественного типа.

Недочёты при выполнении задания могут быть устранены следующим образом: больше практиковаться в написании программ на языке Си.

Подпись студента Минеева С.А