МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский Авиационный Институт»

(Национальный Исследовательский Университет)

Институт №8: «Информационные технологии и прикладная математика»

Кафедра 806: «Вычислительная математика и программирование»

**КУРСОВАЯ РАБОТА №7**

По курсу «Вычислительные системы»

**II** семестр

Тема:

**«Разреженные матрицы»**

**Группа:** М80-106Б-22

**Студент:** Ларченко А.О

**Преподаватели:** Дубинин А.В.

**Оценка: \_\_\_\_\_\_\_\_\_ Подпись: \_\_\_\_\_\_\_\_**

**Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Москва, 2023

# Содержание

[**Содержание 2**](#_5bjxas8qpzef)

[**Введение 3**](#_uyifey50bwp5)

[**Программа 4**](#_9jo81si2g25l)

[Задание 4](#_bfqk89p80hab)

[Немного теории 4](#_5j9qdh2j844w)

[Алгоритм 5](#_3h3ck1r78l1l)

[Описание файлов и функций 5](#_ivl8bapzslky)

[**Тестирование 13**](#_7e16my4hesaw)

[**Заключение 17**](#_be10rxpujcie)

# Введение

7 Курсовой проект направлен на знакомство с разреженными матрицами и их способами представления в памяти компьютера.

# Программа

## Задание

Составить программу на языке Си с функциями для обработки *прямоугольных* разреженных матриц согласно схеме размещения матрицы. Программа должна выполнять следующие действия :

1. Вводить матрицу с клавиатуры и хранить только ненулевые элементы в памяти компьютера
2. Печатать матрицу в представлении памяти компьютера
3. Печатать матрицу в нормальном виде
4. Выполнить преобразование согласно заданию и напечатать матрицу

*Элементы матрицы :* целый тип

*Метод представления матрицы*: 3 вектора

*Доп задание:* 2. Определить максимальный по модулю элемент матрицы и разделить на него все элементы столбца, в котором он находится. Если таких элементов несколько, обработать предпоследний столбец, содержащий такой элемент.

## Немного теории

Разреженная матрица - матрица в которой преимущественно больше нулевых элементов, чем ненулевых, следовательно нам бессмысленно хранить все эти нулевые значения, а если матрица больших размеров, то это уже вопрос памяти! Поэтому для борьбы с этой проблемой используется метод представления такой матрицы в виде 3 векторов(CIP, PI, YE),

* вектор YE хранит все ненулевые элементы матрицы, в порядке их расположения в матрице в порядке обхода(слева-направо и сверху вниз)
* вектор PI - хранит индексы столбцов, в которых встречаются элементы из YE. Отсюда можно заметить, что оба этих вектора имеют одинаковую длину
* вектор CIP - фактически показывает количество ненулевых элементов в строке.

## Алгоритм

Пользователь с клавиатуры вводит размеры матрицы, а потом передаёт саму матрицу на ввод. Программа сразу считывает матрицу в соответствии со способом хранения. Затем последовательно выполняются все перечисленные в задании действия, а результат выводится в терминал.

Также для хранения данных, программа использует структуру вектор.

Сложность программы - О(n\*m), где m и n - размеры матрицы:

* заполнение структуры, печать массива, - все за О(n\*m)
* печать массива во внутреннем представлении и выполнение функции по заданию - О(k), k - количество ненулевых элементов.

В последнем пункте такой сложности помогает добиться вектор, доступ к элементов к которому за О(1).

## Описание файлов и функций

main.h - описание структуры для хранения 3 векторов, а также описание всех функций

#include <stdio.h>

#include <stdbool.h>

#include <stdlib.h>

#include <string.h>

#include "vectorCerclBuf.h"

#define START\_SIZE 50

typedef struct data{

int m, n;

int count;

int k\_cip;

ddl\_vector\_Cercle CIP; //с какого индекса начинаются каждая строчка, типо того

ddl\_vector\_Cercle PI; // столбцы, в которых расположены элементы

ddl\_vector\_Cercle YE; // value

}data;

// void create\_data(data mas, int m, int n);

data create\_data(int m, int n);

void mat\_print(data mas);

void vec\_print(data mas);

void function(data mas);

main.c - реализация всех функций

#include "main.h"

static void zero\_input(ddl\_vector\_Cercle \*v, int l, int r){

for(int i=l; i<r; ++i){

set\_el(v, i, 0);

}

}

static void veprint(ddl\_vector\_Cercle \*v, int l, int r){

for(int i=l; i<r;++i){

printf("%d ", get\_el(v, i));

}

}

void vec\_print(data mas){

printf("\n");

printf(" Vector CIP: ");

veprint(&mas.CIP, 0, mas.k\_cip);

printf("\n");

printf(" Vector PI: ");

veprint(&mas.PI, 0, mas.count);

printf("\n");

printf(" Vector YE: ");

veprint(&mas.YE, 0, mas.count);

printf("\n");

}

void function(data mas){

int mx=1;

int id\_max\_f=0;

int id\_max\_s=0;

for(int i=0; i<mas.count;++i){

int cur\_el=abs(get\_el(&mas.YE, i));

if(cur\_el>mx){

id\_max\_s=get\_el(&mas.PI, i);

// id\_max\_s=id\_max\_f;

mx=cur\_el;

} else if(cur\_el==mx){

if(get\_el(&mas.PI, i)>=id\_max\_f){

if(get\_el(&mas.PI, i)>id\_max\_s){

id\_max\_f=id\_max\_s;

id\_max\_s=get\_el(&mas.PI, i);

} else{

id\_max\_f=get\_el(&mas.PI, i);

}

}

}

}

for(int i=0; i<mas.count;++i){

if(get\_el(&mas.PI, i)==id\_max\_f){

int tmp;

tmp=get\_el(&mas.YE, i);

tmp=tmp/mx;

set\_el(&mas.YE, i, tmp);

}

}

}

void mat\_print(data mas){

int id\_cip=0;

int id\_pi=0;

for (int i=0; i<mas.m;++i){

int c=get\_el(&mas.CIP, id\_cip+1)-get\_el(&mas.CIP, id\_cip);

int cnt\_py=0;

for (int j=0; j<mas.n; ++j){

int point=0;

if(cnt\_py<c){

if(get\_el(&mas.PI, id\_pi)==j){

printf("%d ", get\_el(&mas.YE, id\_pi));

id\_pi++;

cnt\_py++;

} else{

printf("%d ", 0);

}

} else{

printf("%d ", 0);

}

}

printf("\n");

id\_cip++;

}

}

data create\_data(int m, int n){

data mas;

mas.m=m;

mas.n=n;

mas.count=0;

init(&mas.CIP);

init(&mas.PI);

init(&mas.YE);

set\_size(&mas.CIP, START\_SIZE);

set\_size(&mas.PI, START\_SIZE);

set\_size(&mas.YE, START\_SIZE);

mas.k\_cip=0;

zero\_input(&mas.PI, 0, START\_SIZE);

set\_el(&mas.CIP, mas.k\_cip,0);

for(int i=0; i<m;++i){

for(int j=0;j<n;++j){

int tmp;

scanf("%d", &tmp);

if(tmp!=0){

set\_el(&mas.YE, mas.count, tmp);

set\_el(&mas.PI, mas.count, j);

mas.count++;

if(mas.count>=START\_SIZE){

set\_size(&mas.PI, START\_SIZE\*2);

set\_size(&mas.YE, START\_SIZE\*2);

}

}

}

if(mas.k\_cip>=START\_SIZE){

set\_size(&mas.CIP, START\_SIZE\*2);

}

set\_el(&mas.CIP, i+1, mas.count);

mas.k\_cip++;

}

set\_el(&mas.CIP, mas.k\_cip+1, mas.count);

mas.k\_cip++;

return mas;

// printf(" Vector CIP: ");

// veprint(&mas.CIP, 0, mas.k\_cip);

// printf("\n");

// printf(" Vector PI: ");

// veprint(&mas.PI, 0, mas.count);

// printf("\n");

// printf(" Vector YE: ");

// veprint(&mas.YE, 0, mas.count);

}

bool is\_exit(int c){

if(c>='0' && c<='9'){

return false;

}

return true;

}

int main(){

int n,m;

int c;

printf("Enter matrix A(m/n) size, or if you want to exit, press 'q'\n");

printf("m = ");

c=getchar();

while(c!= EOF && c!='q'){

// printf("Enter matrix A(m/n) size, or if you want to exit, press 'q'\n");

// printf("m = ");

// c=getchar();

data mas;

if(is\_exit(c)==false){

ungetc(c, stdin);

scanf("%d", &m);

printf("n = ");

scanf("%d", &n);

mas=create\_data(m, n);

vec\_print(mas);

printf("\n");

mat\_print(mas);

function(mas);

printf("\n");

mat\_print(mas);

}

printf("Enter matrix A(m/n) size, or if you want to exit, press 'q'\n");

printf("m = ");

c=getchar();

}

}

# Тестирование

Первым делом соберем наши файлы.

arsenii@LarchCompu:~/Documents/prog/projects/ones/Cprog/kp7$ make

gcc -c main.c -o main.o

gcc main.o vectorCerclBuf.o -o AL\_kp7

Тесты:

1. Входные данные: 5 5

0 0 1 6 0

0 1 0 0 0

0 0 0 0 0

1 2 0 0 0

0 0 6 0 0

Результат работы программы:

Vector CIP: 0 2 3 3 5 6

Vector PI: 2 3 1 0 1 2

Vector YE: 1 6 1 1 2 6

0 0 1 6 0

0 1 0 0 0

0 0 0 0 0

1 2 0 0 0

0 0 6 0 0

0 0 0 6 0

0 1 0 0 0

0 0 0 0 0

1 2 0 0 0

0 0 1 0 0

1. Входные данные: 1 5

4 4 4 4 4

Результат работы программы:

Vector CIP: 0 5

Vector PI: 0 1 2 3 4

Vector YE: 4 4 4 4 4

4 4 4 4 4

4 4 4 1 4

1. Входные данные 5 5

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

Выходные данные:

Vector CIP: 0 0 0 0 0 0

Vector PI:

Vector YE:

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

# 

# 

# Заключение

Благодаря этой работе я познакомился с разреженными матрицами и научился с ними работать. Считаю, что с заданием справился успешно.