

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 9 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження алгоритмів обходу
масивів»

Варіант 27

Виконав студент

ПІ-13 Пархомчук Ілля Вікторович
(шифр, прізвище, ім'я, по батькові)

Перевірів

Всечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 9

Дослідження алгоритмів обходу масивів

Мета – дослідити алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Варіант 27

№ варіанта	Опис варіанту
27	Задано матрицю дійсних чисел $A[m,n]$. При обході матриці по рядках знайти в ній перший максимальний елемент X і його місцезнаходження. Обміняти знайдене значення X з елементом останнього стовбця

Постановка задачі

Потрібно ініціалізувати двовимірний масив. Потім знайти перший максимальний елемент та його координати і за допомогою них обміняти з елементом останнього стовбця.

Побудова математичної моделі

Таблиця змінних

Змінна	Тип	Ім'я	Призначення
Матриця	Індексований дійсний	A	Проміжні дані, вихідні дані
Кількість рядків в матриці A	Цілочисельний	m	Вхідні дані, проміжні дані
Кількість стовпців в матриці A	Цілочисельний	n	Вхідні дані, проміжні дані
Елемент	Дійсний	X	Проміжні дані, Вихідні дані
Номер рядка X	Цілочисельний	X_m	Проміжні дані
Номер стовпця X	Цілочисельний	X_n	Проміжні дані
Лічильник	Цілочисельний	i	Проміжні дані

Лічильник	Цілочисельний	j	Проміжні дані
Допоміжна змінна знаходження максимального елементу	Дійсний	max	Проміжні дані
Допоміжна змінна для обміну	Дійсний	temp	Проміжні дані

Таблиця аргументів підпрограм

Матриця до друку/опрацювання	Індексований символний	arr	Проміжні дані, вихідні дані
Кількість рядків	Цілочисельний	m	Проміжні дані
Кількість стовпчиків	Цілочисельний	n	Проміжні дані
Номер рядка елемента	Цілочисельний	el_m	Проміжні дані
Номер стовпчика елемента	Цілочисельний	el_n	Проміжні дані

Таблиця підпрограм

Підпрограма	Оголошення	Опис
init_MxN_matrix	init_MxN_matrix (arr[,], m, n)	Ініціалізує матрицю випадковими значеннями
print_MxN_matrix	print_MxN_matrix (arr[,], m, n)	Друкує матрицю
get_first_max_pos	get_first_max_pos (arr[,], m, n, el_m, el_n)	Повертає згенерований одновимірний масив на основі переданого двовимірного
swap_max_and_last_row_elements	swap_max_and_last_row_elements (arr[,], m, n, el_m, el_n)	Друкує одновимірний масив

Таблиця операторів

Оператор	Назва	Синтаксис	Опис
rand()	Рандом	rand()	Повертає випадкове натуральне число

Потрібно заповнити матрицю випадковими дійсними значеннями. Для цього можна використати функцію rand(). Так як rand() повертає випадкове натуральне число, то різниця rand() - rand() точно буде цілим числом(друге випадкове число може бути більше ніж перше). І щоб перейти до дійсних чисел можна отриману різницю поділити на деяке натуральне, відмінне від нуля число(наприклад 100).

Потрібно знайти перший максимальний елемент. Для цього потрібно використати строгу нерівність при порівнянні. Тоді, коли зустрінеться елемент з таким самим значенням умова буде хибною, адже елемент не може бути більшим самого себе.

Далі, знаючи координати елемента поміняти його місцями з останнім елементом в його рядку(таким чином це буде останній стовчик).

Розв'язання

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо ініціалізацію матриці за допомогою підпрограми з використанням арифметичного оператора повторення.

Крок 3. Деталізуємо пошук першого максимального елемента масиву за допомогою підпрограми з використанням арифметичного оператора повторення та умовної форми вибору.

Крок 4. Деталізуємо обмін першого максимального елемента з елементом останнього стовпчика за допомогою підпрограми з використанням арифметичного оператора повторення та умовної форми вибору.

Крок 5. Деталізуємо вивід матриці за допомогою підпрограми з використанням арифметичного оператора повторення.

Псевдокод алгоритму

Основна програма

початок

ввiд m

ввiд n

init_MxN_matrix (A, m, n)

print_MxN_matrix(A, m, n)

get_first_max_pos(A, m, n, X_m, X_n)

X = A[X_m,X_n]

вивiд «First max: »

swap_max_and_last_row_elements(A, m, n, X_m, X_n)

print_MxN_matrix(A, m, n)

кiнець

Пiдпрограми

пiдпрограма init_MxN_matrix(arr[,], m, n)

iнiцiалiзацiя матрицi

все пiдпрограма

пiдпрограма get_first_max_pos (arr[,], m, n, el_m, el_n)

пошук першого максимального елементу

все пiдпрограма

пiдпрограма swap_max_and_last_row_elements (arr[,], m, n, el_m, el_n)

обмiн першого максимального елементу з елементом останнього стовпчика

все пiдпрограма

пiдпрограма print_MxN_matrix (arr[,], m, n)

друк матрицi

все пiдпрограма

Основна програма

початок

ввід m

ввід n

init_MxN_matrix (A, m, n)

print_MxN_matrix(A, m, n)

get_first_max_pos(A, m, n, X_m, X_n)

X := A[X_m,X_n]

вивід «First max: »

swap_max_and_last_row_elements(A, m, n, X_m, X_n)

print_MxN_matrix(A, m, n)

кінець

Підпрограми

підпрограма init_MxN_matrix(arr[,], m, n)

для i **від** 0 **до** m, **з кроком** 1 **повторити**

для j **від** 0 **до** n, **з кроком** 1 **повторити**

arr[i,j] := (rand() – rand())/100

все повторити

все повторити

все підпрограма

підпрограма get_first_max_pos (arr[,], m, n, el_m, el_n)

max := arr[0,0]

для i від 0 до m, з кроком 1 повторити

для j від 0 до n, з кроком 1 повторити

якщо max < arr[i,j]

то

max := arr[i,j]

el_m := i

el_n := j

все якщо

все повторити

все повторити

все підпрограма

підпрограма swap_max_and_last_row_elements (arr[,], m, n, el_m, el_n)

temp := arr[el_m][el_n]

arr[el_m][el_n] := arr[el_m][n - 1]

arr[el_m][n - 1] := temp

все підпрограма

підпрограма print_MxN_matrix (arr[,], m, n)

для i від 0 до m, з кроком 1 повторити

для j від 0 до n, з кроком 1 повторити

вивід arr[i,j]

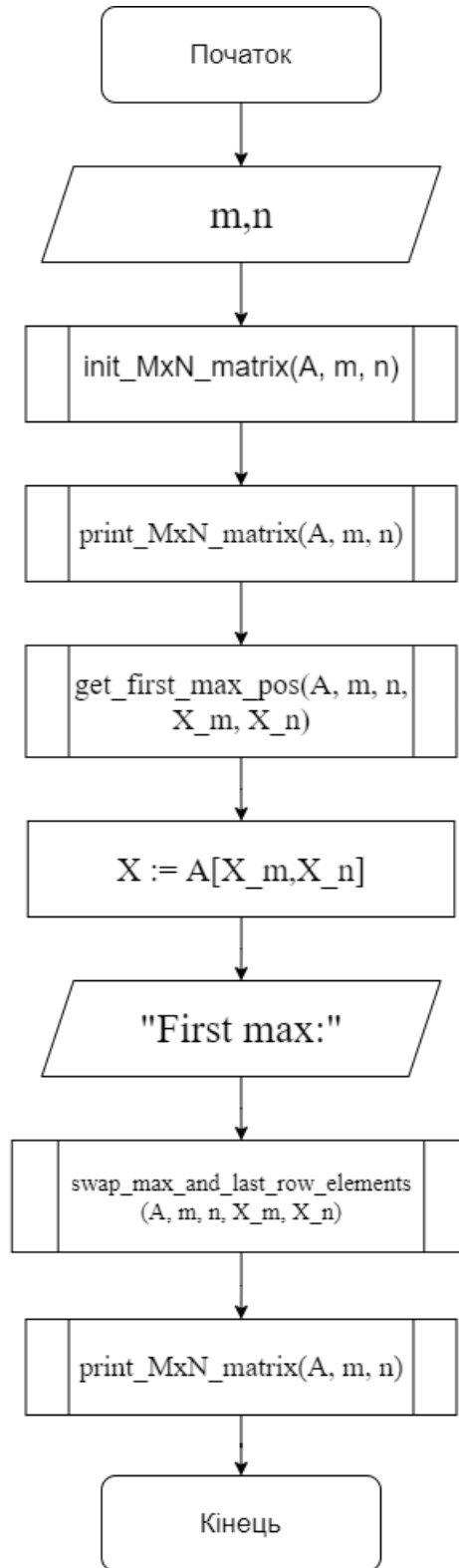
все повторити

все повторити

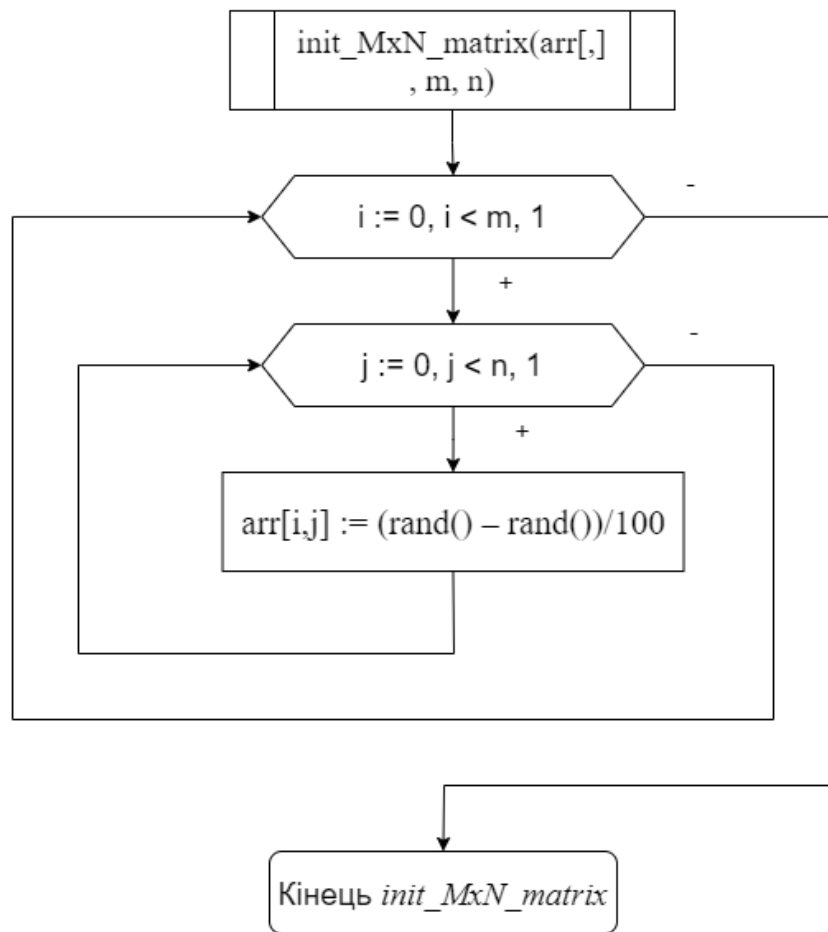
все підпрограма

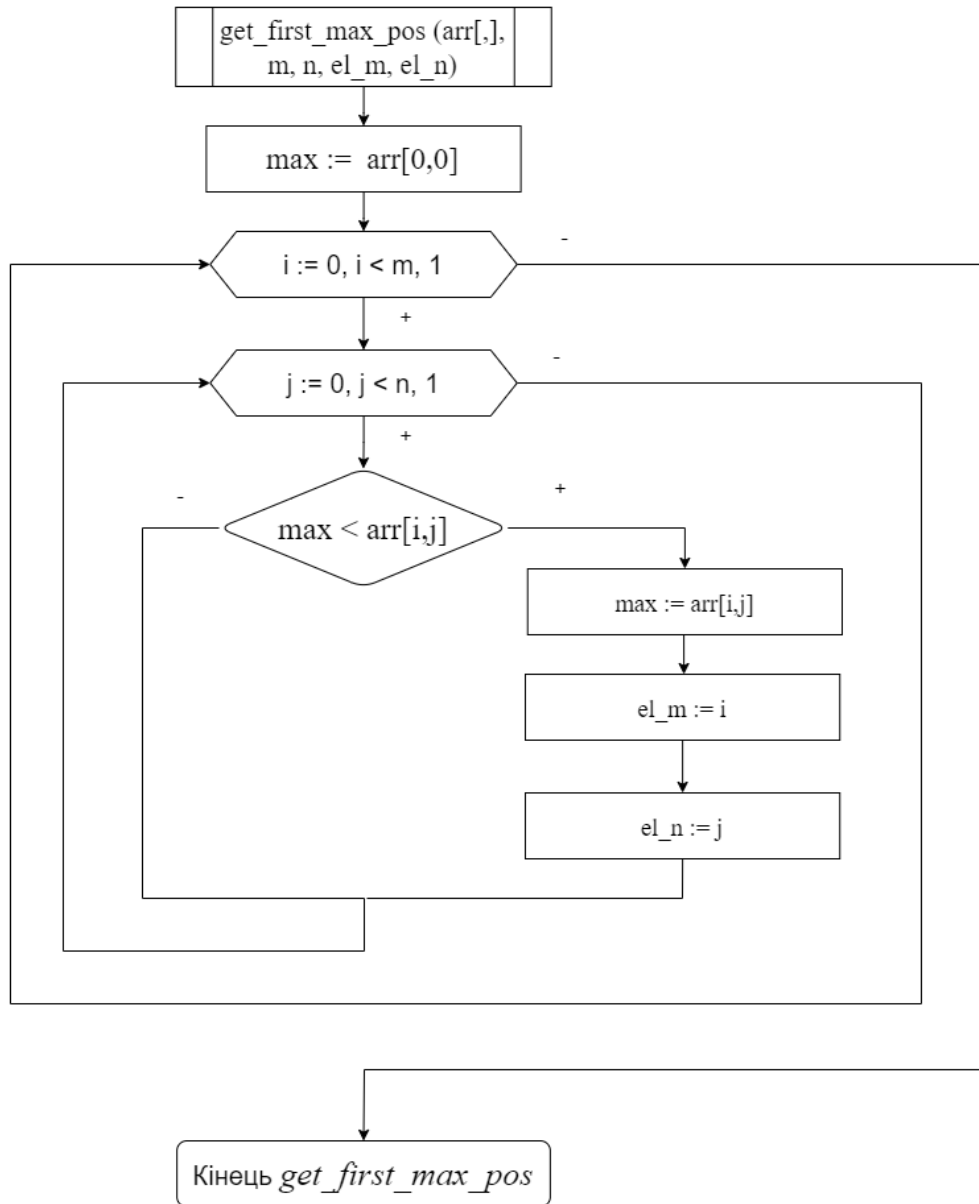
Блок-схема

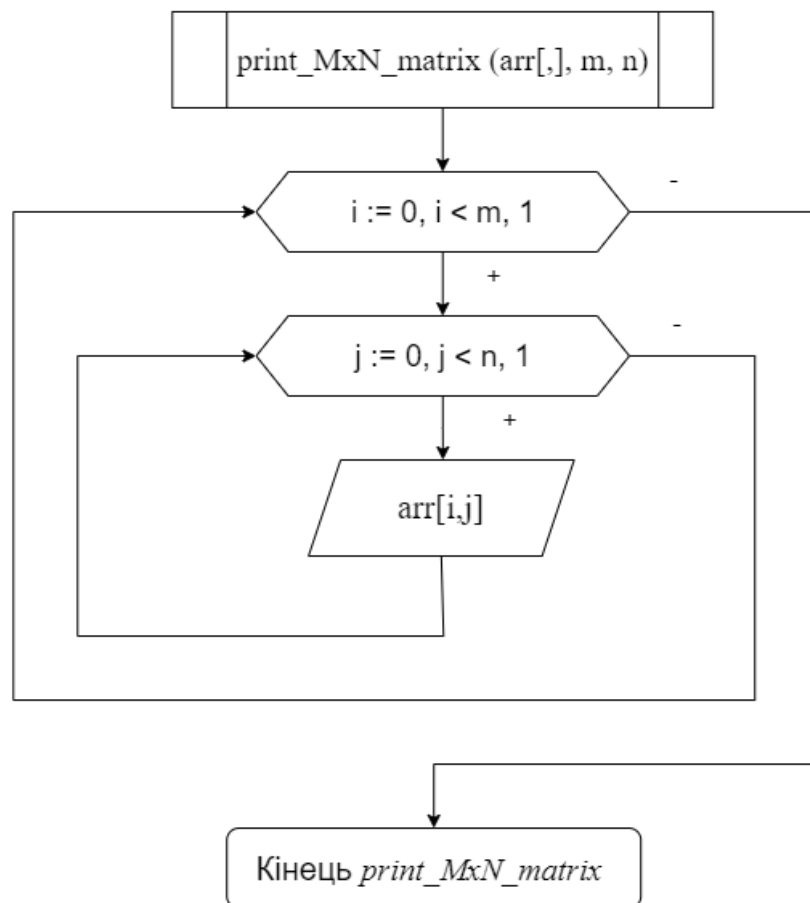
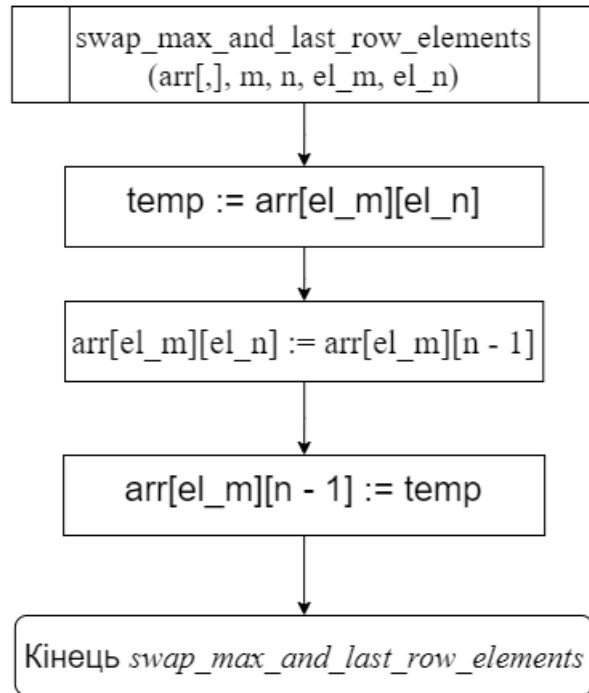
Основна програма:



Підпрограми:







Код програми

```
#include <iostream>
#include <time.h>
#include <random>
#include <iomanip>

using std::cout;
using std::cin;
using std::setw;

void init_MxN_matrix(double** arr, const int m, const int n);
void print_MxN_matrix(double** arr, const int m, const int n);
void get_first_max_pos(double** arr, const int m, const int n, int* el_m, int* el_n);
void swap_max_and_last_row_elements(double** arr, const int m, const int n, int* el_m, int* el_n);

int main()
{
    srand(time(NULL));
    int m, n, X_m, X_n;
    double X;
    cout << "Enter m: "; cin >> m;
    cout << "Enter n: "; cin >> n;
    double** A = new double* [m];
    for (int i = 0; i < m; ++i)
    {
        A[i] = new double[n];
    }
    init_MxN_matrix(A, m, n);
    print_MxN_matrix(A, m, n);
    get_first_max_pos(A, m, n, &X_m, &X_n);
    X = A[X_m][X_n];
    cout << "First max: " << X << '\n';
    swap_max_and_last_row_elements(A, m, n, &X_m, &X_n);
    print_MxN_matrix(A, m, n);

    for (int i = 0; i < m; ++i)
    {
        delete[] A[i];
    }
    delete[] A;
    system("pause");
    return 0;
}

void init_MxN_matrix(double** arr, const int m, const int n)
{
    for (int i = 0; i < m; ++i)
    {
        for (int j = 0; j < n; ++j)
        {
            arr[i][j] = ( rand() - rand() ) / 100.0;
        }
    }
}

void print_MxN_matrix(double** arr, const int m, const int n)
{
    for (int i = 0; i < m; ++i)
    {
        for (int j = 0; j < n; ++j)
        {
            cout << setw(8) << arr[i][j];
        }
        cout << '\n';
    }
    cout << '\n';
}
```

```

void get_first_max_pos(double** arr, const int m, const int n, int* el_m, int* el_n)
{
    double max = arr[0][0];
    for (int i = 0; i < m; ++i)
    {
        for (int j = 0; j < n; ++j)
        {
            if (max < arr[i][j])
            {
                max = arr[i][j];
                *el_m = i;
                *el_n = j;
            }
        }
    }
}

void swap_max_and_last_row_elements(double** arr, const int m, const int n, int* el_m,
int* el_n)
{
    double temp = arr[*el_m][*el_n];
    arr[*el_m][*el_n] = arr[*el_m][n - 1];
    arr[*el_m][n - 1] = temp;
}

```

Випробування алгоритму

```
Enter m: 8
Enter n: 6
-215.66  101.61  -79.25  115.39 -219.75    3.31
-22.15   -48.92  -81.99   83.74  -45.79  -87.16
 18.22  -237.43    2.98  -44.21  -26.03   94.86
-77.69  -211.18   83.64  212.62  225.33  -14.38
-90.92  -171.53   97.39  -17.26  -80.76  -62.84
104.44  -153.96   20.64   76.29  150.22  234.79
-109.8  -316.86 -109.08   -91.7  127.25  -57.02
-30.85   277.5  292.56  230.73  134.89  -37.51

First max: 292.56
-215.66  101.61  -79.25  115.39 -219.75    3.31
-22.15   -48.92  -81.99   83.74  -45.79  -87.16
 18.22  -237.43    2.98  -44.21  -26.03   94.86
-77.69  -211.18   83.64  212.62  225.33  -14.38
-90.92  -171.53   97.39  -17.26  -80.76  -62.84
104.44  -153.96   20.64   76.29  150.22  234.79
-109.8  -316.86 -109.08   -91.7  127.25  -57.02
-30.85   277.5  -37.51  230.73  134.89  292.56
```

```
Enter m: 4
Enter n: 10
-65.59  -13.47   79.23 -133.88   76.78  -40.84  -46.58   84.48  209.95   9.32
  2.2   114.49 -149.65 -108.92  234.25   44.4  211.84 -110.13  197.57   76.5
258.21 -298.81 -298.52 -184.5  128.44   94.61   30.01   14.43  138.31  154.14
-251.37 224.54  -54.14  -74.81 -150.18 -114.09  174.38  -51.3  -112.47  -90.51

First max: 258.21
-65.59  -13.47   79.23 -133.88   76.78  -40.84  -46.58   84.48  209.95   9.32
  2.2   114.49 -149.65 -108.92  234.25   44.4  211.84 -110.13  197.57   76.5
154.14 -298.81 -298.52 -184.5  128.44   94.61   30.01   14.43  138.31  258.21
-251.37 224.54  -54.14  -74.81 -150.18 -114.09  174.38  -51.3  -112.47  -90.51
```

Висновки

На лабораторній роботі я дослідив алгоритми обходу масивів, набув практичних навичок використання цих алгоритмів під час складання програмних специфікацій. Спочаку я ініціалізував матрицю випадковими дійсними числами, потім знайшов координати першого максимального елементу. Потім поміняв його місцями з елементом останнього стовпчика. Також створив підпрограми для виводу матриці. Мною був написаний псевдокод, створено блок-схеми та проведено тестування алгоритму.