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 30

Виконав студент ІП-13 Симотюк Денис Андрійович
(шифр, прізвище, ім'я, по батькові)

Перевірив Вєчерковська 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 30

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом.

2. Ініціювання змінної, що описана в п.1 даного завдання.

3. Обчислення змінної, що описана в п.1, згідно з варіантом.

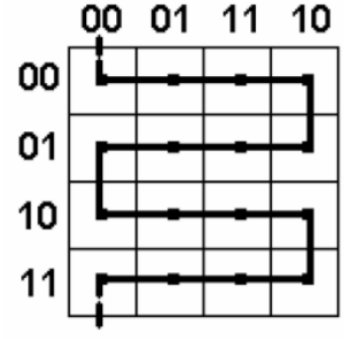
30	Задано матрицю дійсних чисел A[m,n]. При обході матриці по рядках знайти в ній останній мінімальний елемент X і його місцезнаходження. Порівняти значення X з середньоарифметичним значенням елементів під головною діагоналлю.

Постановка задачі

Для вирішення задачі створюється масив, який заповнюється дійсними числами. Для збереження мінімального елементу і його індексу створюються 3 змінні. Для обчислення середнього арифметичного створимо три змінні, одній присвоюється значення суми елементів, що нижче головної діагоналі, а другій – кількість цих елементів, третя ж буде результатом ділення перших двох. Після цього вона буде порівнюватися з раніше збереженим мінімальним елементом і виведеться відповідне повідомлення.

Побудова математичної моделі

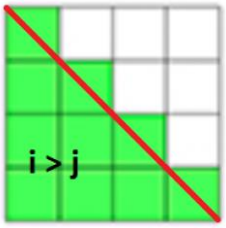
Обхід масиву по рядках – алгоритм, при якому рядки з парним індексом обходяться з початку, а з непарним – з кінця:



Середнє арифметичне – відношення суми елементів до їх кількості:

$$\frac{a1 + a2 + \ldots + an}{n}$$

Елементи матриці, що знаходяться нижче головної діагоналі – такі, рядок яких більший за стовпчик:



Побудуємо таблицю змінних:

Змінна	Тип	Призначення
Масив **arr(m, n)**	Дійсний	Вхідні дані
Кількість рядків **m**	Цілий	Вхідні дані
Кількість стовпців **n**	Цілий	Вхідні дані
Лічильник циклу **i**	Цілий	Проміжні дані, лічильник
Лічильник циклу **j**	Цілий	Проміжні дані, лічильник
Останнє мінімальне значення **min**	Дійсний	Вихідні дані
Рядок min **minRow**	Цілий	Вихідні дані
Стовпець min **minCol**	Цілий	Вихідні дані
Чисельник середнього арифметичного **sum**	Дійсний	Проміжні дані
Знаменник середнього арифметичного **quant**	Цілий	Проміжні дані
Середнє арифметичне **avg**	Дійсний	Проміжні дані
Пошук мінімального елементу **Search_Min**	Функція	Проміжні дані
Обчислення середнього арифметичного Count_Avg	Функція	Проміжні дані

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо знаходження min, minRow, minCol.

Крок 3. Деталізуємо знаходження avg.

Крок 4. Деталізуємо функцію Search_Min.

Крок 5. Деталізуємо функцію Count_Avg.

Крок 6. Деталізуємо порівняння min і avg.

Псевдокод алгоритму

Початок

Введення arr

min = Search_Min(arr, m, n)

avg = Count_Avg(arr, m, n);

Якщо min < avg

то

Виведення «min < avg»

інакше

Виведення «min > avg»

Все якщо

Виведення min, minRow, minCol, avg

Кінець

Функція Search_Min(arr, m, n)

min = arr[0, 0]

minRow = 0

x`

Повторити для i від 0 до m з кроком 1

Якщо i % 2 == 0

то

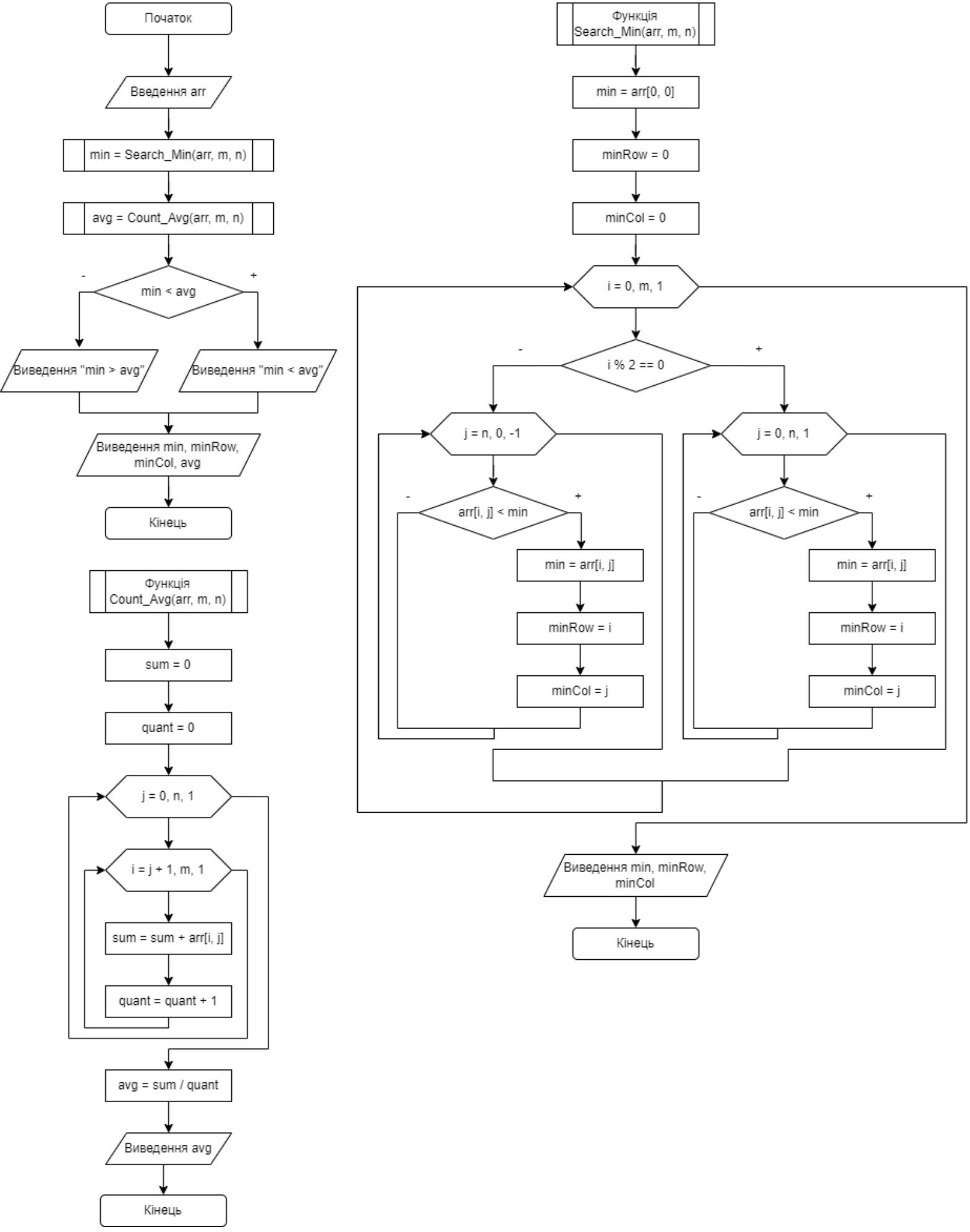
Повторити для j від 0 до n з кроком 1

Якщо arr[i, j] < min

то

```
                                    min = arr[i, j]
                                    minRow = i
                                    minCol = j
                            все якщо
                        все повторити
                    інакше
                        Повторити для j від n до 0 з кроком -1
                            Якщо arr[i, j] < min
                                то
                                    min = arr[i, j]
                                    minRow = i
                                    minCol = j
                            все якщо
                        все повторити
                все якщо
            все повторити
            Виведення min, minRow, minCol
Все функція
Функція Count_Avg(arr, m, n)
        sum = 0
        quant = 0
        Повторити для j від 0 до n з кроком 1
                Повторити для i від j+1 до m з кроком 1
                    sum = sum + arr[i, j]
                    quant = quant + 1
                все повторити
        все повторити
        avg = sum / quant
        Виведення avg
все функція
```

Блок-схема алгоритму



Код програми (C++)

```cpp

#include <iostream>
#include <iomanip>
using namespace std;

double** Create_Matrix(int rows, int cols);
void Init_Matrix(double** arr, int rows, int cols);
double Search_Min(double** arr, int rows, int cols, int* rowPtr, int* colPtr);
double Count_Avg(double** arr, int rows, int cols);

int main()
{
    int m, n, minRow = 0, minCol = 0;
    int* rowPtr = &minRow, *colPtr = &minCol;
    double min, avg;
    double** arr;
    srand(int(time(NULL)));

    cout << "Enter number of rows: ";
    cin >> m;
    cout << "Enter number of columns: ";
    cin >> n;
    cout << endl;

    arr = Create_Matrix(m, n);
    Init_Matrix(arr, m, n);
    for (int i = 0; i < m; i++) { ... }

    min = Search_Min(arr, m, n, rowPtr, colPtr);

    rowPtr = NULL;
    colPtr = NULL;
    delete rowPtr;
    delete colPtr;

    cout << "Minimal element: " << min << endl;
    cout << "Its position: (" << minRow + 1 << ", " << minCol + 1 << ")" << endl;

```

```
45
46      avg = Count_Avg(arr, m, n);
47      cout << "Average = " << avg << endl;
48
49      if (min < avg)
50          cout << "min < avg" << endl;
51      else
52          cout << "min > avg";
53
54  }
55
56  double** Create_Matrix(int rows, int cols)
57  {
58      double** arr = new double*[rows];
59      for (int i = 0; i < rows; i++)
60          arr[i] = new double[cols];
61      return arr;
62  }
63
64  void Init_Matrix(double** arr, int rows, int cols)
65  {
66      for (int i = 0; i < rows; i++)
67          for (int j = 0; j < cols; j++)
68              arr[i][j] = double(rand() % 40) - 20;
69
70  }
71
72  double Search_Min(double** arr, int rows, int cols, int *rowPtr, int *colPtr)
73  {
74      double min = arr[0][0];
75      for (int i = 0; i < rows; i++)
76      {
77          if (i % 2 == 0)
78          {
79              for (int j = 0; j < cols; j++)
80                  if (arr[i][j] <= min)
81                  {
82                      min = arr[i][j];
```

```
82                      min = arr[i][j];
83                      *rowPtr = i;
84                      *colPtr = j;
85                  }
86              }
87              else
88              {
89                  for (int j = cols - 1; j >=0; j--)
90                      if (arr[i][j] <= min)
91                      {
92                          min = arr[i][j];
93                          *rowPtr = i;
94                          *colPtr = j;
95                      }
96              }
97          }
98          return min;
99      }
100
101     double Count_Avg(double** arr, int rows, int cols)
102     {
103         double avg, sum = 0, quant = 0;
104         for (int j = 0; j < cols; j++)
105             for (int i = j + 1; i < rows; i++)
106             {
107                 sum += arr[i][j];
108                 quant++;
109             }
110         avg = sum / quant;
111         return avg;
112     }
```

Тестування програми

```
Enter number of rows: 4
Enter number of columns: 4

  -15  -20   -6    6
   -1   12    3   -6
    8   14  -19  -13
   -3   16    1    1

Minimal element: -20
Its position: (1, 2)
Average = 5.83333
min < avg

C:\Users\Денис\source\repos\lab 9 asd\Debug\lab 9 asd.exe (процесс 3952) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно…
```

Висновки

В даній роботі я дослідив алгоритми обходу масивів, набув практичних навичок використання цих алгоритмів під час складання алгоритмів та написання програм на мові програмування C++.