

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 лінійних алгоритмів»

Варіант 34

Виконав студент ІП-12 Шоман Данило Володимирович
(шифр, прізвище, ім'я, по батькові)

Перевірів

(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 9

Дослідження рекурсивних алгоритмів

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

Варіант 34

Умова:

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

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом (табл. 1).
2. Ініціювання змінної, що описана в п.1 даного завдання.
3. Обчислення змінної, що описана в п.1, згідно з варіантом (табл. 1).

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

Постановка задачі: Введемо змінні індексованого типу: arr (двовимірний), і змінні n – розмір масиву, $minimal$ – значення мінімального елементу, pos_min_col – значення стовпця мінімального значення, pos_min_row – значення рядка мінімального значення, sum , $counter$ – допоміжні змінні для визначення $serAr$, $serAr$ – середнє арифметичне елементів під побічною діагоналлю. Результатом розв’язку є порівняння змінної $minimal$ з $serAr$. Для визначення результату потребується лише ввести значення змінної n . Оператори: $rand()$ – генерація випадкових значень.

Математична модель:

Змінна	Ім'я	Тип	Призначення
Розміри Arr	n	Цілий	Початкове дане
Двовимірний масив	arr	Індексований	Проміжне значення

Положення мінімального елемента	pos_min_col, pos_min_row	Цілий	Проміжне значення
Значення мінімального елемента	minimal	Дійсний	Проміжне значення
Лічильник	i, j	Цілий	Проміжне значення
Допоміжні змінні для розрахунку середнього арифметичного	sum, counter	Дійсний	Проміжне значення
Середнє арифметичне елементів під побічною діагоналлю	serAr	Дійсний	Проміжне значення

План:

Крок 1. Визначимо основні дії.

Крок 2. Деталізація заповнення масиву, *arr* елементами

Крок 4. Деталізація пошуку мінімального елемента

Крок 5. Деталізація порівняння середнього арифметичного і мінімального елемента

Псевдокод:

крок 1

початок

Заповнення масиву *arr* елементами

Пошук мінімального елемента

Порівняння середнього арифметичного і мінімального елемента

кінець

крок 2

початок

введення n

arr = **виклик**

підпрограми

input(n);

виведення arr

Пошук

мінімального

елемента

Порівняння

середнього

арифметичного і
мінімального
елемента
кінець

крок 3

початок

введення n

arr = **виклик підпрограми** input(n);

виведення arr

minimal = **виклик підпрограми** searchPoint(arr, n);

Порівняння середнього арифметичного і мінімального
елемента

кінець

крок 4

початок

введення n

arr = **виклик підпрограми** input(n);

виведення arr

minimal = **виклик підпрограми** searchPoint(arr, n);

виклик підпрограми compare(arr, n, minimal);

Підпрограма input(n)

arr[n]

повторити

для i від 0 до row, i++

для j від 0 до col, j++

arr[i] [j] = rand()

все повторити

все повторити

повернути arr

кінець підпрограма

Підпрограма searchPoint(arr, n)

minimal = 151;

pos_min_col = 0, pos_min_row = 0;

dir = 1;

```

повторити
  для i від 0 до n, i++
    повторити
      якщо dir > 0
        то:
          для j від 0 до n, j++
            повторити
              якщо Arr[i][j] <= minimal
                то:
                  minimal = arr[i][j];
                  pos_min_col = j;
                  pos_min_row = i;
                все якщо
              все повторити
            все якщо
          якщо dir < 0
            то:
              для j від n-1 до 0, j--
                повторити
                  якщо Arr[i][j] <= minimal
                    то:
                      minimal = arr[i][j];
                      pos_min_col = j;
                      pos_min_row = i;
                    все якщо
                  все повторити
                все якщо
              dir = dir * (-1)
            все повторити
          вивести minimal, pos_min_col, pos_min_row
        повернути minimal
      кінець підпрограма

```

Підпрограма *compare*(*arr*, *n*, *minimal*)

sum = 0, *counter* = 0;

повторити

для *i* від 0 до *row*, *i*++

для *j* від 0 до *col*, *j*++

якщо *i* + *j* + 1 > *n*

то:

sum += Arr[i][j];

counter += 1;

все якщо

все повторити

все повторити

serAr = sum / counter;

якщо minimal < serAr

то:

вивести «Найменший елемент менше»

або якщо minimal = serAr

вивести «Рівні»

інакше

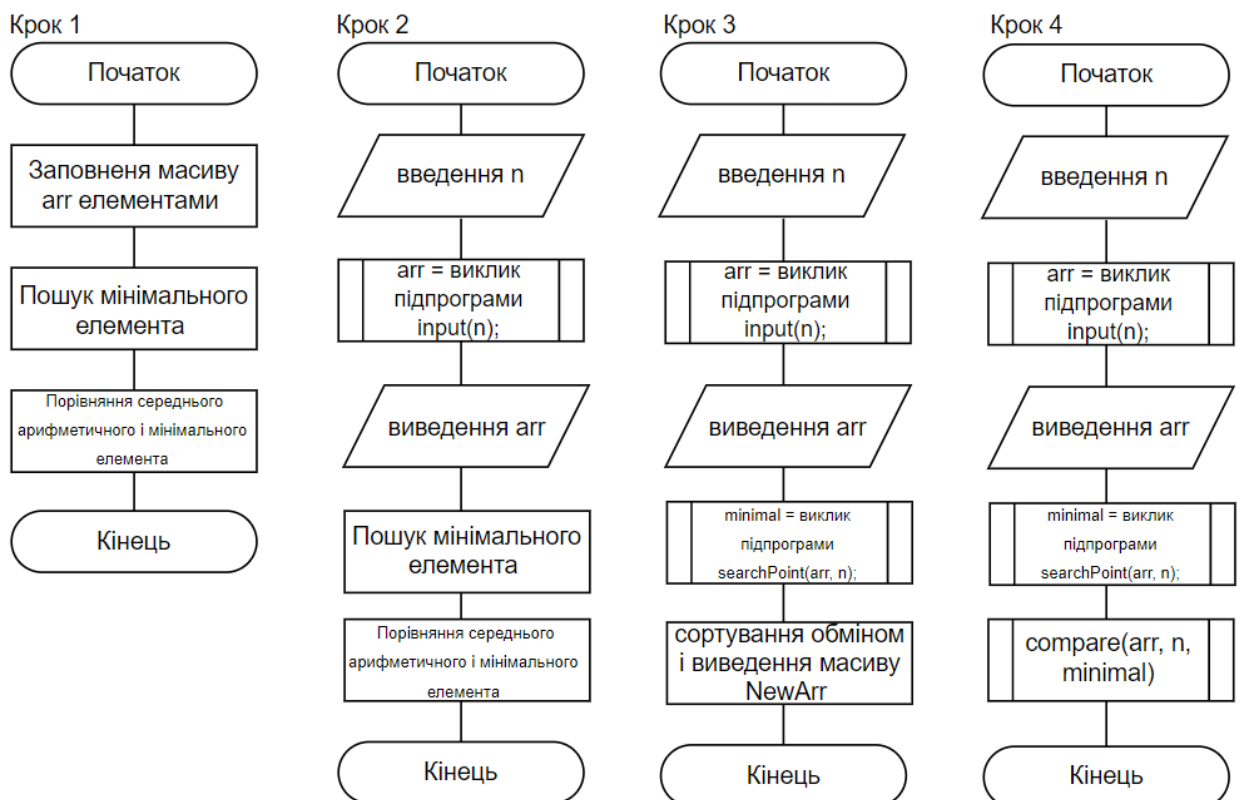
вивести «Середнє арифметичне менше»

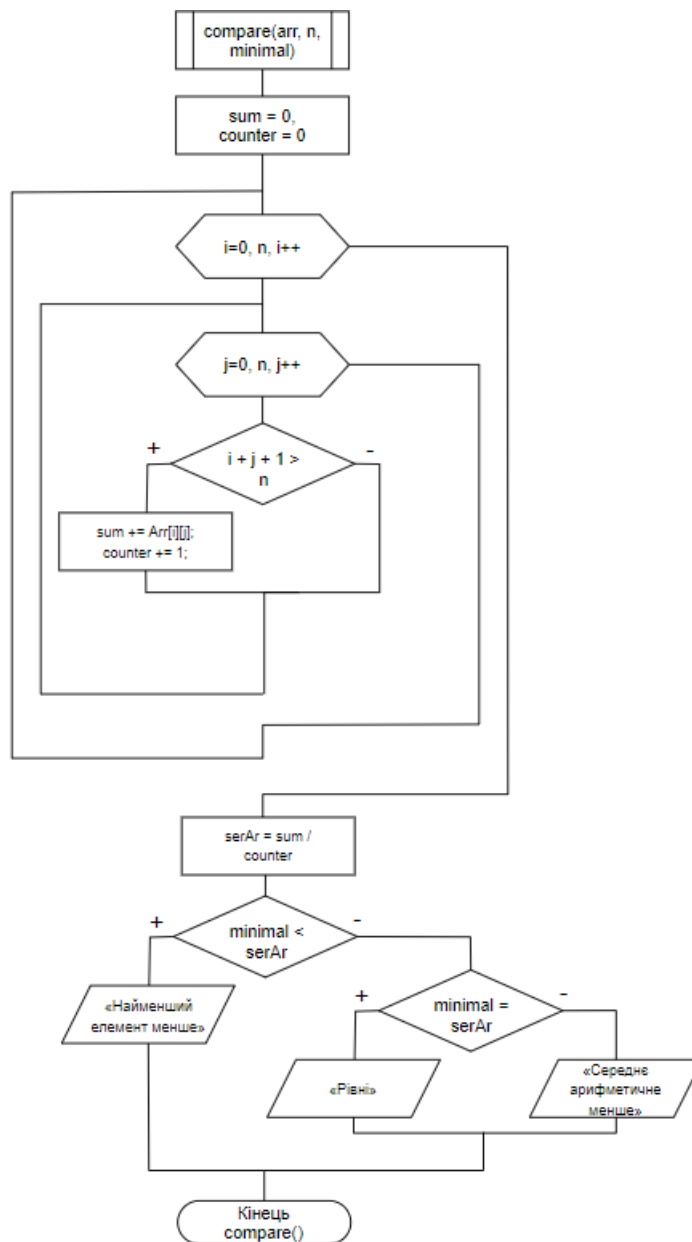
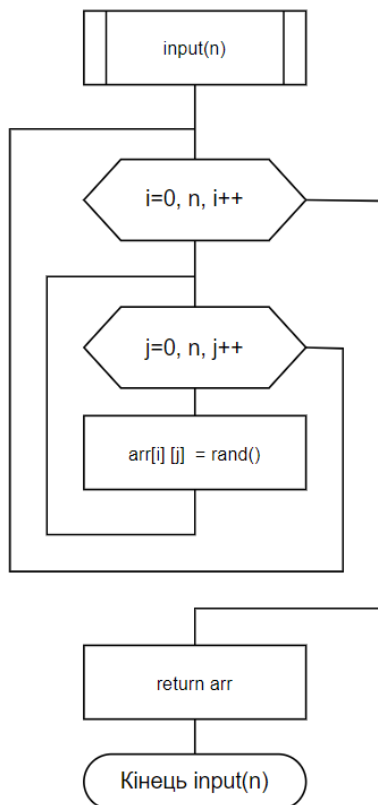
все якщо

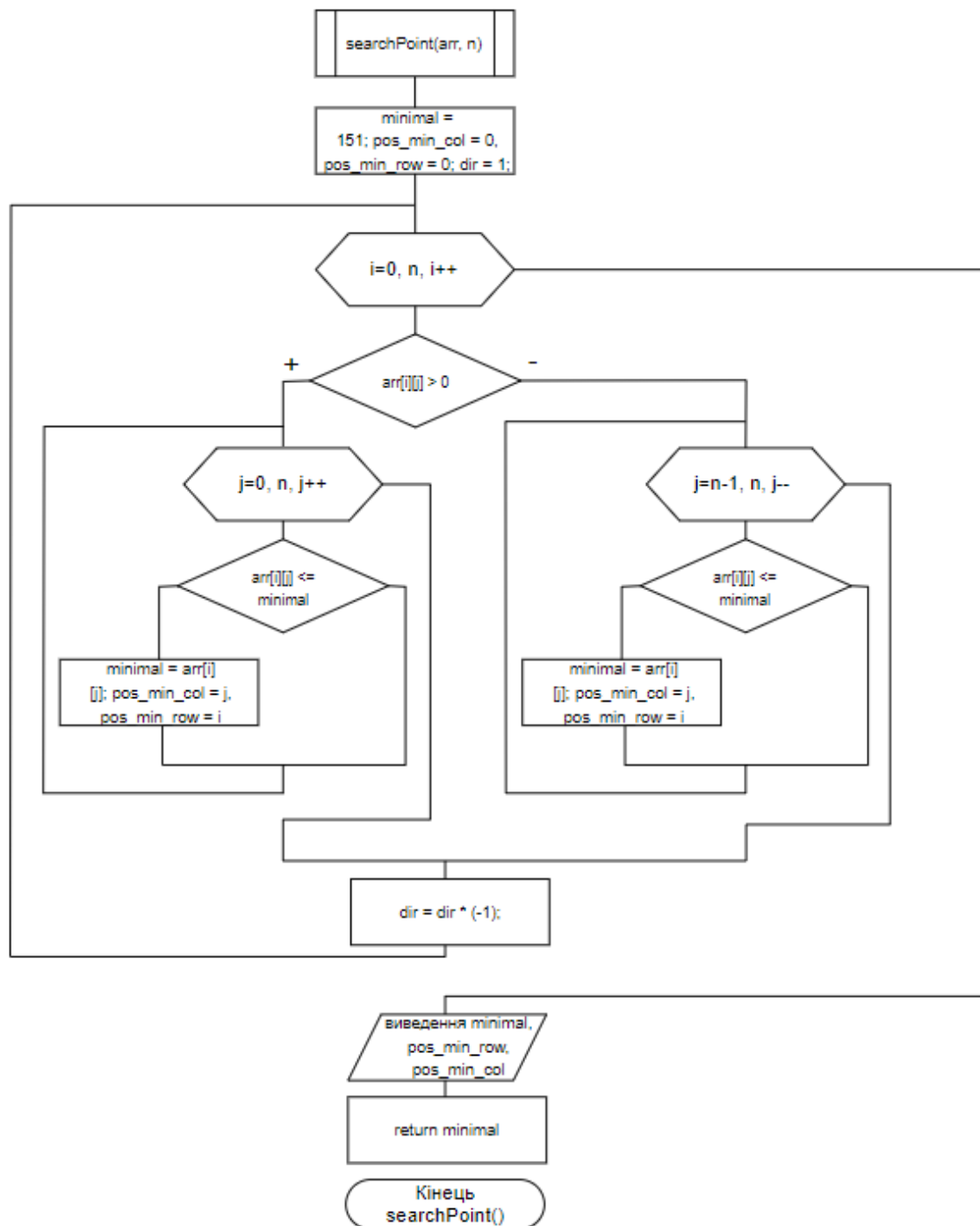
кінець підпрограма

кінець

Блок-схема:







Код:

```

#include <iostream>
#include <time.h>
#include <iomanip>
#include <stdio.h>
using namespace std;

float** input(int);
void outputArray(float**, int);
void deleteArray(float**, int);
float searchPoint(float**, int);
void compare(float**, int, float);

```



```

int main() {
    srand(time(NULL));
    setlocale(LC_ALL, "rus");
    int n;
    cout << "Введіть розмір квадратної матриці: ";
    cin >> n;
    float** arr;
    arr = input(n);
    outputArray(arr, n);
    float minimal = searchPoint(arr, n);
    compare(arr, n, minimal);
    deleteArray(arr, n);
}

float** input(int n) {
    float** arr = new float* [n];
    for (int i = 0; i < n; i++)
        arr[i] = new float[n];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] = float(rand()) / RAND_MAX * 200 - 50;;
        }
    }
    return arr;
}

void outputArray(float** Arr, int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++)
            cout << setw(7) << fixed << setprecision(1) << Arr[i][j];
        cout << endl;
    }
}

float searchPoint(float** Arr, int n) {
    float minimal = 151;
    int pos_min_col = 0, pos_min_row = 0;
    int dir = 1;
    for (int i = 0; i < n; i++) {
        if (dir > 0) {
            for (int j = 0; j < n; j++) {
                if (Arr[i][j] <= minimal) {
                    minimal = Arr[i][j];
                    pos_min_col = j;
                    pos_min_row = i;
                }
            }
        }
        else {
            for (int j = n - 1; j >= 0; j--) {
                if (Arr[i][j] <= minimal) {
                    minimal = Arr[i][j];
                    pos_min_col = j;
                    pos_min_row = i;
                }
            }
            dir = dir * (-1);
        }
    }
    printf("Найменший елемент - %3.1f, його положення - [%d, %d]", minimal,
pos_min_row+1, pos_min_col+1);
    return minimal;
}

```

```

    }

    void compare(float** Arr, int n, float minimal) {
        float sum = 0, counter = 0;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (i + j + 1 > n) {
                    sum += Arr[i][j];
                    counter += 1;
                }
            }
        }
        float serAr = sum / counter;
        cout << endl;
        if (minimal < serAr) {
            printf("Найменший елемент (%3.1f) менший нїж середнє арифметичне (%3.1f)",
minimal, serAr);
        }
        else if (minimal == serAr) {
            printf("Найменший елемент (%3.1f) такий самий, як 1 середнє арифметичне
(%3.1f)", minimal, serAr);
        }
        else {
            printf("Найменший елемент (%3.1f) бїльший нїж середнє арифметичне (%3.1f)",
minimal, serAr);
        }
    }

    void deleteArray(float** Arr, int n) {
        for (int i = 0; i < n; i++)
            delete[] Arr[i];
        delete[] Arr;
    }
}

```

Випробування:

Microsoft Visual Studio Debug Console

```

Введїть розмїр квадратної матрицї?: 5
-11.1  123.1  -23.5  46.7  50.0
-16.2  95.6   85.9  41.6  -15.3
55.3   52.1   12.6  47.7  -16.2
-23.6  121.1  43.7  134.1  5.5
111.9  84.4   30.7  84.6   91.5
Найменший елемент - -23,6, його положення - [4, 1]
Найменший елемент (-23,6) менший нїж середнє арифметичне (49,1)

```

Microsoft Visual Studio Debug Console

```

Введїть розмїр квадратної матрицї?: 4
-9.5   -28.7   99.3   -3.7
86.4   101.2   15.4    56.5
103.9  -17.3   -16.9   73.6
22.6   127.0   19.8    42.3
Найменший елемент - -28,7, його положення - [1, 2]
Найменший елемент (-28,7) менший нїж середнє арифметичне (50,4)

```

```
Введіть розмір квадратної матриці?: 2
-7.4 -9.1
-34.1 111.5
Найменший елемент - -34,1, його положення - [2, 1]
Найменший елемент (-34,1) менший ніж середнє арифметичне (111,5)
```

Висновки: На цій лабораторній я досліджував алгоритми обходу масивів.

Мій варіант лабораторної включав роботу з операторами `rand()` – генерування випадкових чисел, роботу з двомірними масивами: їх формування, використання алгоритмів обходу по рядках тощо, пошук мінімального значення, побудову логічних дій розгалуження, циклів, написання підпрограм, деталізацію різних частин псевдокоду і блоксхем. Також я практикувався в умінні оформлювати лабораторну роботу, а саме: титульний аркуш, математичну модель, псевдокод алгоритму, блок схему алгоритму, випробування алгоритму, висновки.