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 № 8 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження алгоритмів пошуку та сортування»

Варіант 5

Виконав студент ІП-11 Гіжицький Даниїл Олександрович

(шифр, прізвище, ім'я, по батькові)

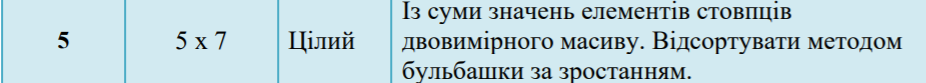
Перевірив Мартинова Оксана Петрівна

( прізвище, ім'я, по батькові)

**Мета** – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

**Варіант 5**

**Завдання:**

 **Розв’язок:**

**1 Постановка задачі**

**Результатом виконання задачі є відсортований за методом вставки масив сум негативних елементів рядків матриці.**

**2 Побудова математичної моделі**

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім'я** | **Призначення** |
| Матриця | Двомірний масив дійсних чисел [7 x 5] | matr | Вхідні дані |
| Масив сум | Масив дійсних чисел [7] | arr | Проміжні дані,  Вихідні дані |

**Створимо функцію для створення матриці з віпадковими значеннями(**rand\_matrix**), функція повертає матрицю.**

**Створимо функцію для визначення головної діагоналі матриці(**sum\_col**), функція повертає масив.**

**Створюємо функцію для сортування масиву методом бульбашки (**sort**), алгоритм:**

На кожному кроці алгоритму ми вибираємо один з елементів вхідних даних і вставляємо його на потрібну позицію у вже відсортованому списку доти, доки набір вхідних даних не буде вичерпано.

**Розв’язання**

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Деталізуємо **функцію (**sum\_col**)**

Крок 2. Деталізуємо **функцію (**sort**)**

Крок 3. Деталізуємо основну програму

**Псевдокод**

*Крок 1*

*Функція* sum\_col ***(list mat)***

**Початок**

Ініціалізація масиву arr

**Повторити** для i від 0 до 7

Ініціалізація int sum = 0

**Повторити** для j від 0 до 5

sum += matr[j][i]

**Кінець повторити**

arr[i] = sum

**Кінець повторити**

Виведення arr

**Кінець**

*Крок 2*

*Функція* sort ***(list arr)***

**Початок**

Ініціалізація bool swaped = 1

**Повторити** для i від 0 до 7

swaped = 0

**Повторити** для j від 1 до 7

**Якщо** arr[j-1] > arr[j]

swap(arr[j-1], arr[j])

swaped = 1

**Кінець повторити**

**Кінець повторити**

**Кінець**

*Крок 3*

*Основна програма*

**Початок**

Введення matr

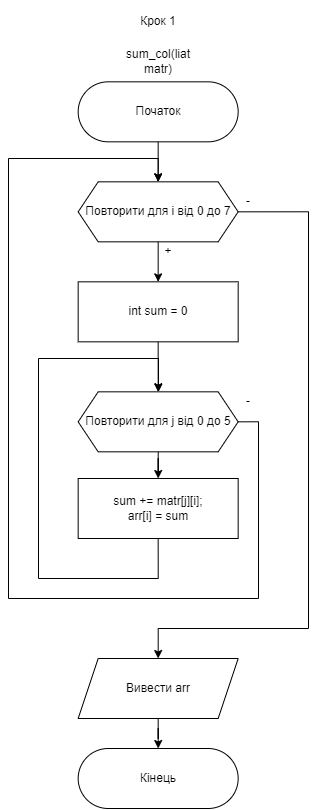
arr = sum\_col(matr)

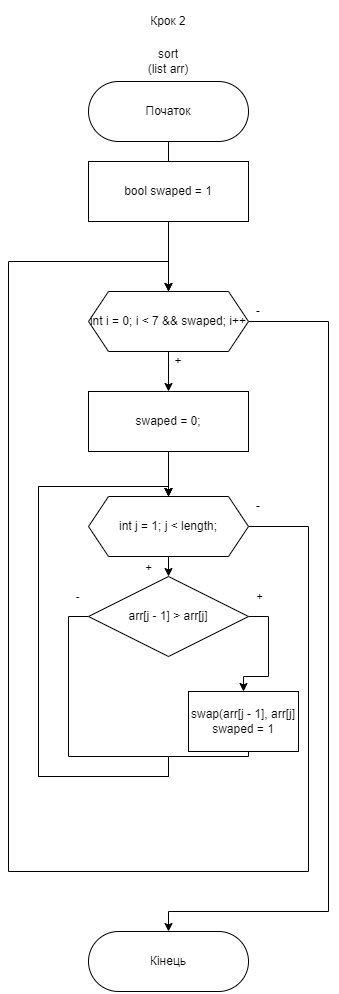
sort (arr)

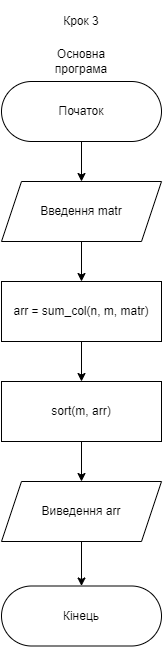
Виведення arr

**Кінець**

**Блок схема**



****

****

**Код програми (с++)**

#include<ctime>

#include <iostream>

using namespace std;

int\*\* rand\_matrix(int n, int m) {

int\*\* mat = new int\* [n];

srand(time(NULL));

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

int\* arr = new int[m];

for (int j = 0; j < m; j++)

arr[j] = rand() % 21 - 10;

mat[i] = arr;

}

return mat;

}

void print\_matrix(int n, int m, int\*\* mat) {

cout << "Matrix:\n";

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

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

cout << mat[i][j] << ' ';

}

cout << "\n";

}

cout << "\n";

}

void print\_array(int length, int\* arr) {

cout << "Array:\n";

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

cout << arr[i] << ' ';

}

cout << '\n';

}

int\* sum\_col(int n, int m, int\*\* matr) {

int\* arr = new int[m];

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

int sum = 0;

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

sum += matr[j][i];

arr[i] = sum;

}

return arr;

}

void sort(int length, int\* arr) {

bool swaped = 1;

for (int i = 0; i < length && swaped; i++)

{

swaped = 0;

for (int j = 1; j < length; j++) {

if (arr[j - 1] > arr[j]) {

swap(arr[j - 1], arr[j]);

swaped = 1;

}

}

}

}

int main()

{

const int n = 5, m = 7;

int\*\* matr = rand\_matrix(n, m);

print\_matrix(n, m, matr);

int\* arr = sum\_col(n, m, matr);

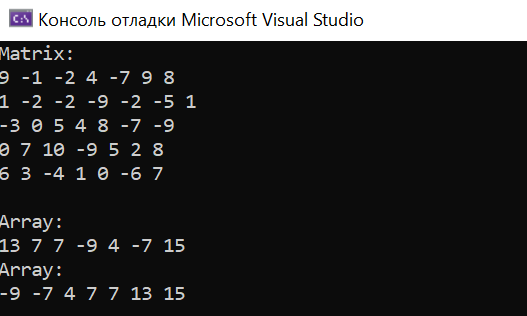
print\_array(m, arr);

sort(m, arr);

print\_array(m, arr);

}

**Тестування**



**Висновок**

Ми дослідили алгоритми пошуку та сортування і набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій, та в результаті отримали відсортований масив значень.