Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

Кафедра ІПЗ

**Лабораторна робота № 6**

**«Методи сортування. Сортування в масивах. Основні алгоритми реалізації»**

Виконав

ст. гр. ІП-22-1

Микитій Ю.М.

Перевірив

доцент Процюк В.Р.

Івано-Франківськ

2023

**Мета:** ознайомитися з основними методами сортування, а також дослідження алгоритмів сортування та вивчення способів їх реалізації за допомогою мови програмування С++

**Умова:**

1. Самостійно створити одновимірний масив з 4 значень цілих чисел:
   1. Побудувати у середовищі Draw.io таблично-графічну схему роботи мотоду сортування бульбашкою(за зростанням)
   2. Побудувати у середовищі таблично-графічну схему сортування за зростанням згідно з методом вставок
2. Заповнити одновимірний цілочисельний масив А(30) з допомогою генератора випадкових чисел (від 0 до 100). Підготувати окремі відповідні програми реалізації вказаних методів для сортування за зростанням отриманого масиву А.

**Розв’язок**

#include <iostream>

#include <cstdlib> // Для використання функції rand()

#include <ctime> // Для використання функції time()

#include <string>

using namespace std;

string foo4(int arr[], int N);

string foo30(int arr[], int N);

void InsertSorter(int a[], int n);

void main()

{

// для генерації випадкових чисел на основі поточного часу

std::srand(static\_cast<unsigned int>(std::time(nullptr)));

//1.-----------------------------

int arr[] = { 98, 48, 45, 11, 7, 3};

for (int i = 0; i < 4; i++)

{

cout << arr[i] << " ";

}

cout << endl;

foo4(arr, 4); //викликаємо функцію сортування

for (int i = 0; i < 4; i++)

{

cout << arr[i] << " ";

}

cout << endl;

cout << "----------------------------" << endl;

//2.

const int arraySize = 30;

int myArray[arraySize];

// Заповнення масиву випадковими числами від 0 до 100

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

myArray[i] = rand() % 101; // Генеруємо випадкове число від 0 до 100

}

// Виведення початкового масиву на екран

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

cout << myArray[i] << " ";

}

cout << endl;

foo30(myArray, 30); //викликаємо функцію сортування

// Виведення масиву на екран

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

cout << myArray[i] << " ";

}

cout << endl;

cout << "----------------------------" << endl;

//3.

int arr3[] = { 22, 19, 9, 8, 11, 14 };

int size = sizeof(arr3) / sizeof(arr[0]);

InsertSorter(arr3, size);

// Print the sorted array

std::cout << "Array3:" << std::endl;

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

cout << arr3[i] << " ";

}

}

string foo4(int \*arr, int N) {

if (N!=4)

{

return "Error!";

}

else {

for (int i = 0; i < N - 1; i++) {

for (int j = 0; j < N - i - 1; j++)

{

if (arr[j] > arr[j + 1])

{

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

return "GOOD!";

}

}

string foo30(int \*arr2, int N) {

if (N != 30)

{

return "Error!";

}

else {

for (int i = 0; i < N - 1; i++) {

for (int j = 0; j < N - i - 1; j++)

{

if (arr2[j] > arr2[j + 1])

{

int temp = arr2[j];

arr2[j] = arr2[j + 1];

arr2[j + 1] = temp;

}

}

}

return "GOOD!";

}

}

void InsertSorter(int arr[], int n) {

int temp, iPrev;

for (int i = 1; i < n; i++) {

temp = arr[i];

iPrev = i - 1;

std::cout << "Step " << i << "\n- compere element:" << std::endl;

// Порівнюємо та вставляємо елемент на вірне місце

while (iPrev >= 0 && arr[iPrev] > temp) {

arr[iPrev + 1] = arr[iPrev];

arr[iPrev] = temp;

iPrev--;

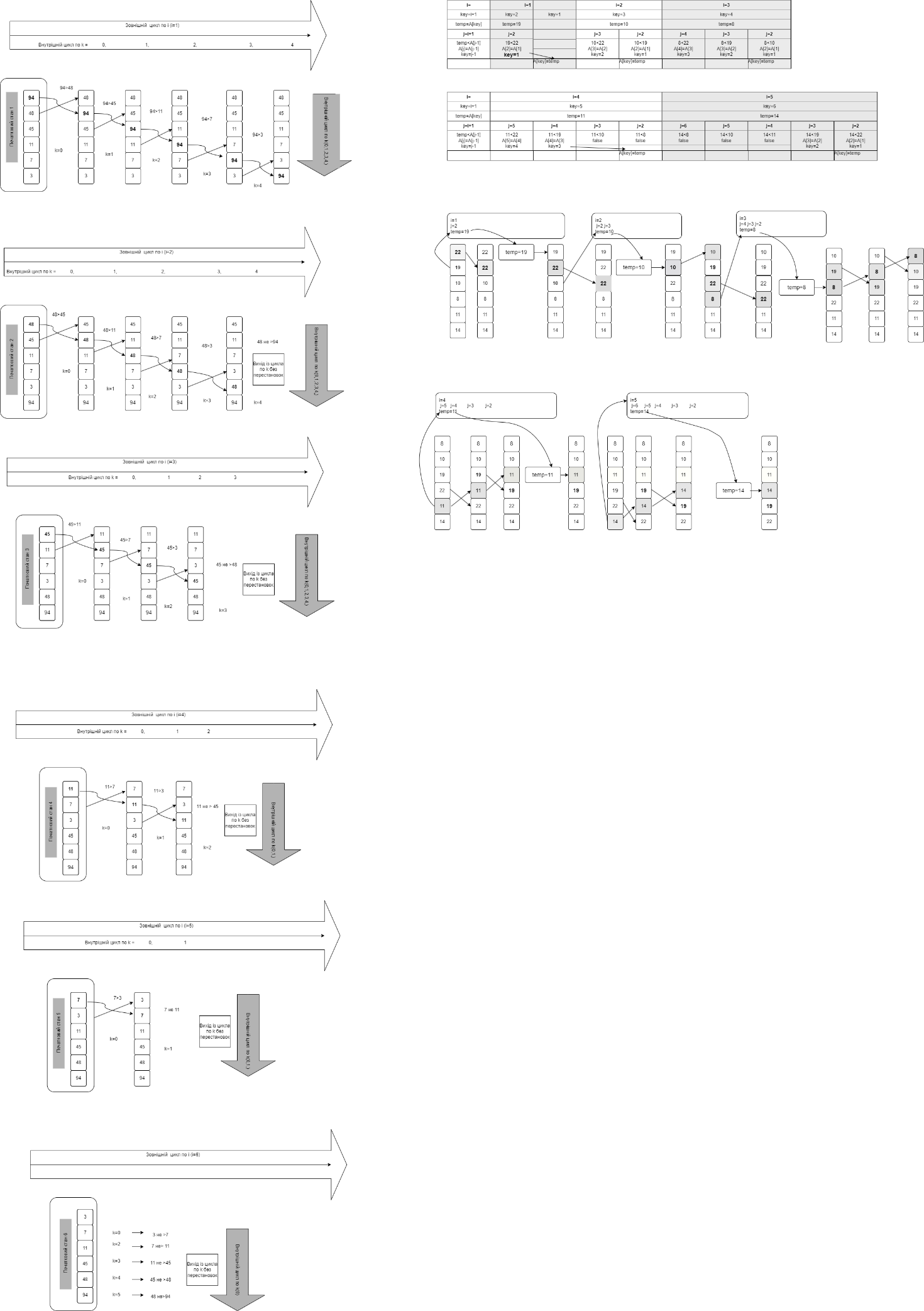
}

}

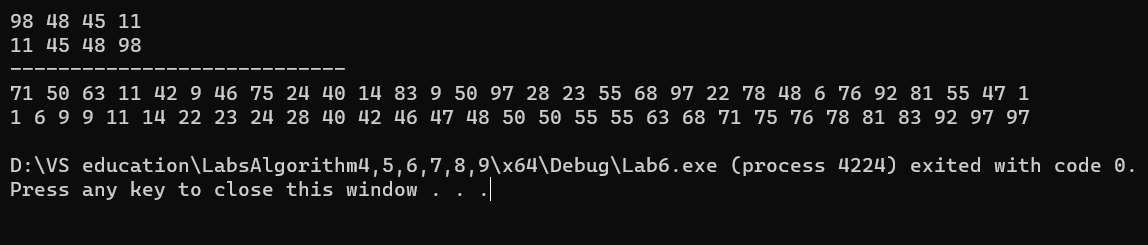
}

Блок-схеми

Сортування бульбашкою Сортування вставкою



Результат



Висновок: на лабораторній роботі, я ознайомився з основними методами сортування, а також дослідив алгоритми сортування та вивчення способів їх реалізації за допомогою мови програмування С++