Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

Кафедра ІПЗ

**Лабораторна робота № 8**

**«Сортування методом Хоара, Злиття»**

Виконав

ст. гр. ІП-22-1

Микитій Ю.М.

Перевірив

доцент Процюк В.Р.

Івано-Франківськ

2023

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

**Завдання:**

1. Заповнити масив цілочисельними випадковими числами в діапазоні (10-100).
2. Розмір масиву N обчислюється як ціла частина від значення виразу N = 20 + 0.6 \* K, де K – номер студента в журналі.
3. Відсортувати масив за методом бульбашки.
4. Знайти ключ, що знаходиться в середині лівої частини масиву. Студентам з парними номерами варіантів використовувати метод бінарногопошуку

**Розв’язок**

#include <iostream>

#include <vector>

// Функція для обміну значень двох елементів

void swap(int& a, int& b) {

int temp = a;

a = b;

b = temp;

}

// Функція для розділення масиву для швидкого сортування

int partition(std::vector<int>& arr, int low, int high) {

int pivot = arr[high];

int i = low - 1;

for (int j = low; j < high; j++) {

if (arr[j] < pivot) {

i++;

swap(arr[i], arr[j]);

}

}

swap(arr[i + 1], arr[high]);

return i + 1;

}

// Основна функція швидкого сортування

void quickSort(std::vector<int>& arr, int low, int high) {

if (low < high) {

int pivotIndex = partition(arr, low, high);

quickSort(arr, low, pivotIndex - 1);

quickSort(arr, pivotIndex + 1, high);

}

}

// Функція для злиття двох відсортованих підмасивів

void merge(std::vector<int>& arr, int left, int middle, int right) {

int n1 = middle - left + 1;

int n2 = right - middle;

// Тимчасові масиви для зберігання двох підмасивів

std::vector<int> leftArray(n1);

std::vector<int> rightArray(n2);

// Копіюємо дані в тимчасові масиви leftArray та rightArray

for (int i = 0; i < n1; i++)

leftArray[i] = arr[left + i];

for (int j = 0; j < n2; j++)

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

// Злиття тимчасових масивів у вихідний масив arr

int i = 0;

int j = 0;

int k = left;

while (i < n1 && j < n2) {

if (leftArray[i] <= rightArray[j]) {

arr[k] = leftArray[i];

i++;

}

else {

arr[k] = rightArray[j];

j++;

}

k++;

}

// Копіюємо залишок leftArray, якщо такий є

while (i < n1) {

arr[k] = leftArray[i];

i++;

k++;

}

// Копіюємо залишок rightArray, якщо такий є

while (j < n2) {

arr[k] = rightArray[j];

j++;

k++;

}

}

// Основна функція злиття сортування

void mergeSort(std::vector<int>& arr, int left, int right) {

if (left < right) {

// Знаходимо середину масиву

int middle = left + (right - left) / 2;

// Сортуємо першу та другу половини

mergeSort(arr, left, middle);

mergeSort(arr, middle + 1, right);

// Злиття відсортованих частин

merge(arr, left, middle, right);

}

}

int main() {

// Приклад використання швидкого сортування

std::cout << "----------1:\n";

std::vector<int> array = { 12, 8, 9, 11, 6, 2 };

int n = array.size();

std::cout << "Original array: ";

for (int num : array) {

std::cout << num << " ";

}

std::cout << std::endl;

quickSort(array, 0, n - 1);

std::cout << "Sorted array: ";

for (int num : array) {

std::cout << num << " ";

}

std::cout << std::endl;

// Приклад використання злиття сортування

std::cout << "\n----------2:\n";

std::vector<int> array2 = { 12, 8, 9, 6, 2 };

int n2 = array2.size();

std::cout << "Original array: ";

for (int num : array2) {

std::cout << num << " ";

}

std::cout << std::endl;

mergeSort(array2, 0, n2 - 1);

std::cout << "Sorted array: ";

for (int num : array2) {

std::cout << num << " ";

}

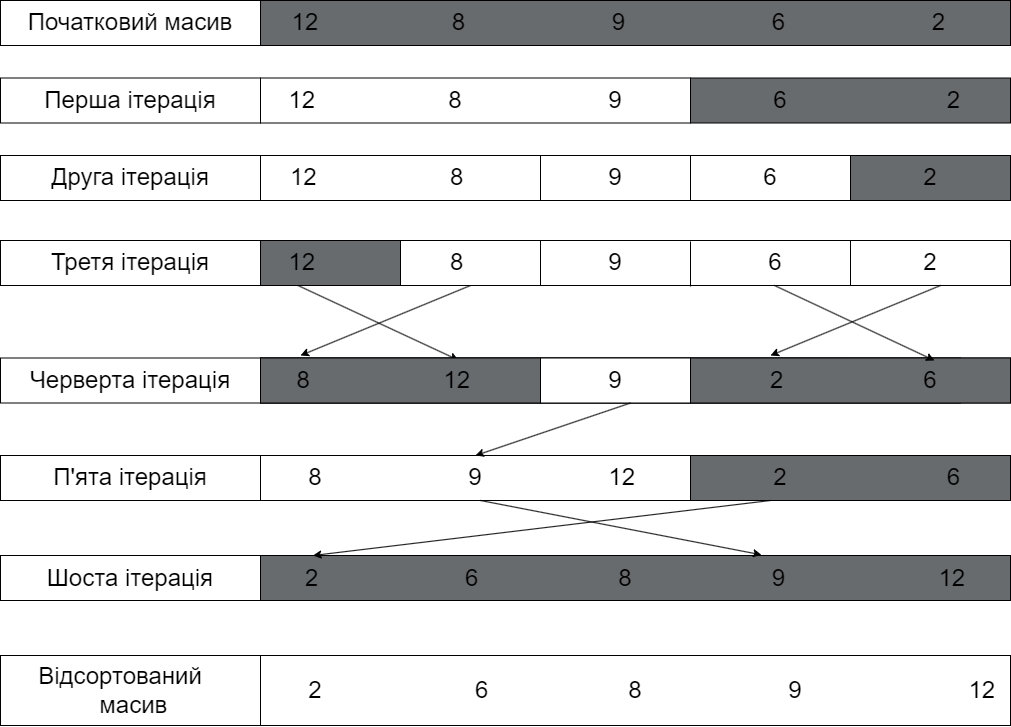
std::cout << std::endl;

return 0;

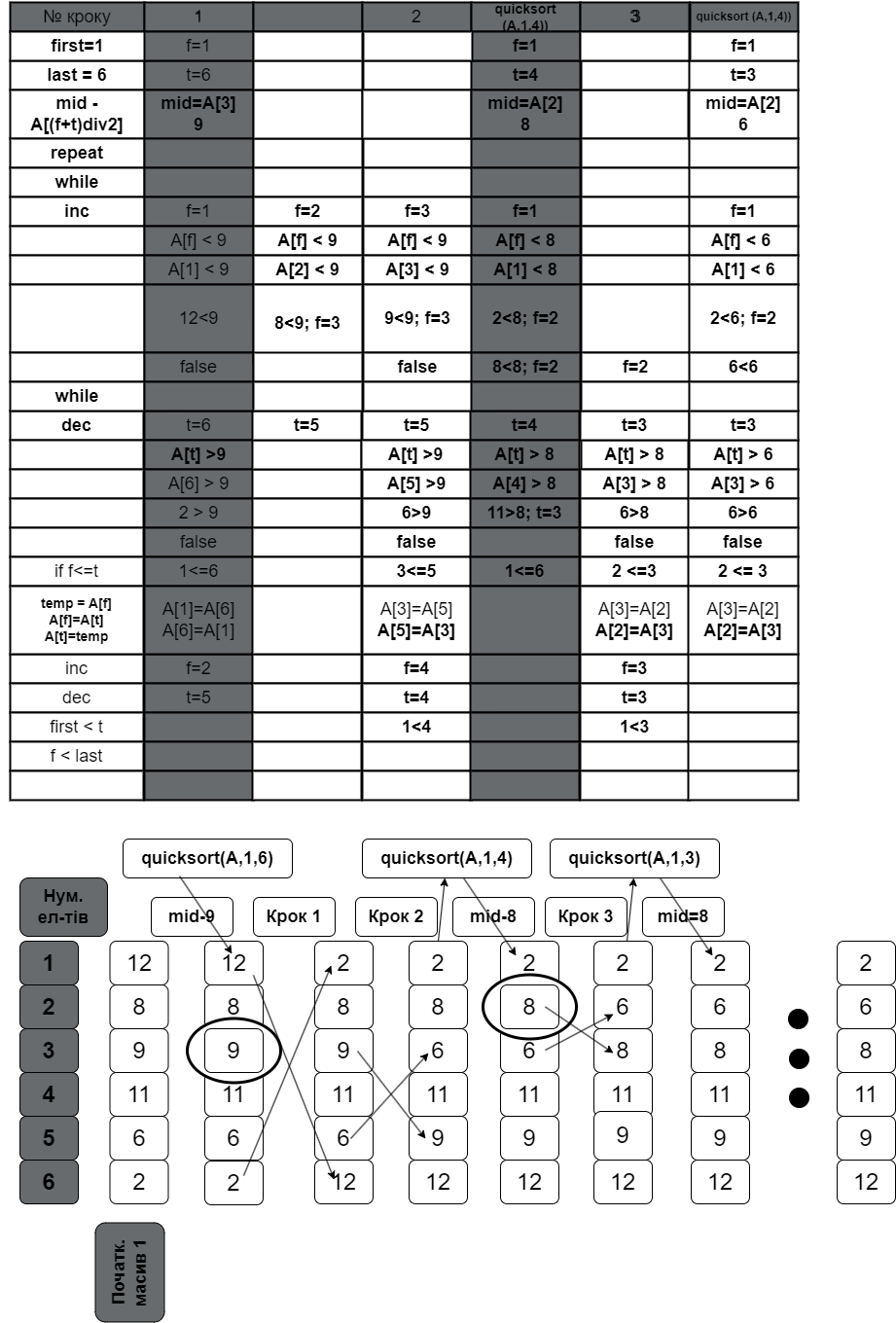
}

**Блок-схеми**

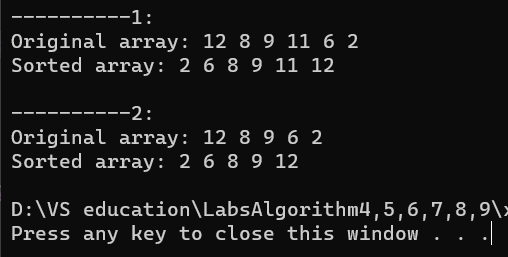
Сортування Злиттям(mergesort)



Сортування методом Хоара(quicksort)



Результат



Висновок: на лабораторній роботі, я **з**робити програму з масивом і відсортував його методами Хоара та Злиття