Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

Кафедра ІПЗ

**Лабораторна робота № 8**

**«Сортування методом Хоара і Злиття »**

Виконав

ст. гр. ІП-22-1

Курчій С.В.

Перевірив

доцент Процюк В.Р.

Івано-Франківськ

2023

**Мета:** ознайомлення 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 шести цілих чисел:

2. Побудувати у середовищі Draw.io таблично-графічну схему роботи методу сортування методом Шелла( за зростанням)

**Розв’язок**

**Метод Хоара:**

#include <iostream>

void printArray(int arr[], int size)

{

    for (int i = 0; i < size; ++i)

    {

        std::cout << arr[i] << " ";

    }

    std::cout << std::endl;

}

int partition(int arr[], int left, int right)

{

    int pivot = arr[(left + right) / 2];

    while (left <= right)

    {

        while (arr[left] < pivot)

        {

            left++;

        }

        while (arr[right] > pivot)

        {

            right--;

        }

        if (left <= right)

        {

            std::swap(arr[left], arr[right]);

            left++;

            right--;

        }

    }

    return left;

}

void quickSortHoare(int arr[], int start, int end)

{

    if (start < end)

    {

        int rightStart = partition(arr, start, end);

        quickSortHoare(arr, start, rightStart - 1);

        quickSortHoare(arr, rightStart, end);

    }

}

int main()

{

    const int size = 6;

    int arr[size] = {12, 7, 10, 5, 2, 4};

    std::cout << "Original array: ";

    printArray(arr, size);

    quickSortHoare(arr, 0, size - 1);

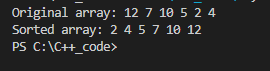
    std::cout << "Sorted array: ";

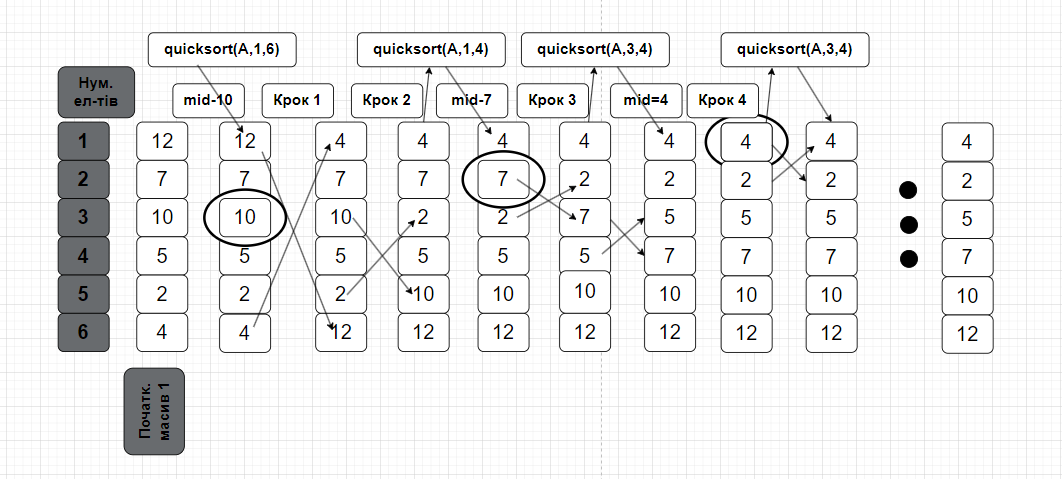
    printArray(arr, size);

    return 0;

}

**Результат**





**Розв’язок**

**Метод Злиття:**

#include <iostream>

#include <vector>

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

{

    int n1 = mid - left + 1;

    int n2 = right - mid;

    std::vector<int> L(n1), R(n2);

    for (int i = 0; i < n1; i++)

        L[i] = arr[left + i];

    for (int j = 0; j < n2; j++)

        R[j] = arr[mid + 1 + j];

    int i = 0, j = 0, k = left;

    while (i < n1 && j < n2)

    {

        if (L[i] <= R[j])

        {

            arr[k] = L[i];

            i++;

        }

        else

        {

            arr[k] = R[j];

            j++;

        }

        k++;

    }

    while (i < n1)

    {

        arr[k] = L[i];

        i++;

        k++;

    }

    while (j < n2)

    {

        arr[k] = R[j];

        j++;

        k++;

    }

}

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

{

    if (left < right)

    {

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

        mergeSort(arr, left, mid);

        mergeSort(arr, mid + 1, right);

        merge(arr, left, mid, right);

    }

}

int main()

{

    std::vector<int> arr = {12, 7, 10, 5, 2, 4};

    std::cout << "Original array: ";

    for (int num : arr)

    {

        std::cout << num << " ";

    }

    mergeSort(arr, 0, arr.size() - 1);

    std::cout << "\nSorted array: ";

    for (int num : arr)

    {

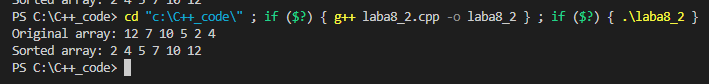
        std::cout << num << " ";

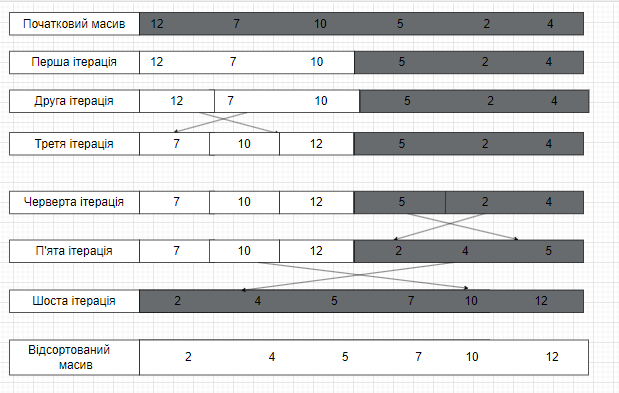
    }

    return 0;

}

**Результат**

****

****