Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

Кафедра ІПЗ

**Лабораторна робота № 9**

**«Двійковий і інтерполяційний методи пошуку »**

Виконав

ст. гр. ІП-22-1

Курчій С.В.

Перевірив

доцент Процюк В.Р.

Івано-Франківськ

2023

**Мета:** вивчення та практичне застосування двійкового та інтерполяційного методів пошуку, а також розвиток навичок реалізації та порівняння алгоритмів у контексті оптимізації пошукових процесів.

**Умова:**

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\* К, де К номер студента в журналі. К = 12

3. Відсортувати масив за методом бульбашки.

4. Знайти ключ, що знаходиться в середині лівої частини масиву.

5. Студентам з парними номерами варіантів використовувати метод бінарного пошуку, а з непарними метод інтерполяційного пошуку.

**Розв’язок**

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <algorithm>

using namespace std;

// Функція для виведення масиву на екран

void printArray(int arr[], int size)

{

    for (int i = 0; i < size; i++)

    {

        cout << arr[i] << " ";

    }

    cout << endl;

}

// Функція для сортування масиву за методом бульбашки

void bubbleSort(int arr[], int size)

{

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

    {

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

        {

            if (arr[j] > arr[j + 1])

            {

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

            }

        }

    }

}

// Функція для знаходження ключа в середині лівої частини масиву

int findMiddleLeftKey(int arr[], int size)

{

    return arr[size / 4];

}

// Функція для бінарного пошуку ключа

int binarySearch(int arr[], int size, int key)

{

    int left = 0, right = size - 1;

    while (left <= right)

    {

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

        if (arr[mid] == key)

        {

            return mid; // Знайдено ключ

        }

        else if (arr[mid] < key)

        {

            left = mid + 1; // Пошук у правій частині

        }

        else

        {

            right = mid - 1; // Пошук у лівій частині

        }

    }

    return -1; // Ключ не знайдено

}

int main()

{

    const int K = 12;                       // Номер студента

    int N = static\_cast<int>(20 - 0.6 \* K); // Обчислення розміру масиву

    const int RANGE\_MIN = 10;

    const int RANGE\_MAX = 100;

    // Ініціалізація генератора випадкових чисел

    srand(time(nullptr));

    // Створення та заповнення масиву випадковими числами

    int arr[N];

    for (int i = 0; i < N; i++)

    {

        arr[i] = rand() % (RANGE\_MAX - RANGE\_MIN + 1) + RANGE\_MIN;

    }

    cout << "Initial array: ";

    printArray(arr, N);

    // Сортування масиву за методом бульбашки

    bubbleSort(arr, N);

    cout << "Sorted array: ";

    printArray(arr, N);

    // Знаходження ключа в середині лівої частини масиву

    int middleLeftKey = findMiddleLeftKey(arr, N);

    cout << "The key is in the middle of the left part: " << middleLeftKey << endl;

    // Бінарний пошук ключа

    int searchResult = binarySearch(arr, N, middleLeftKey);

    if (searchResult != -1)

    {

        cout << "The key is found in the position " << searchResult << endl;

    }

    else

    {

        cout << "Key not found" << endl;

    }

    return 0;

}

**Результат**

