МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ УПРАВЛІННЯ

ЗВІТ

З ЛАБОРАТОРНОЇ РОБОТИ №4

ЗА ТЕМОЮ “Алгоритми сортування”

Виконав студент

групи КН-221д

Габорець

Перевірив

Солонська С.В.

Харків 2021

ЛАБОРАТОРНА РОБОТА №4

Тема: АЛГОРИТМИ СОРТУВАННЯ

Мета роботи: познайомитися алгоритмами сортування та бінарним пошуком.

Завдання :

Розробити програму, яка читає з клавіатури числа N, M (1 < N, M < 256); послідовність N ключів (цілих або дійсних чисел в залежності від варіанту завдання); послідовність M ключів. Програма зберігає першу послідовність до масиву та виконує сортування. Потім програма виводить відсортовану послідовність на екран та виконує бінарний пошук кожного елементу другої послідовністі x: для кожного x повідомити, чи є він у першій послідовністі, а якщо є, то на якому місці.

Варіант завдання:

Сортування злиттям



Код:

HEADER.H:

#pragma once

#include<iostream>

void print\_arr(int\* arr, int size)

{

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

{

std::cout << arr[i] << "\t";

}

std::cout << "\n";

}

//слив массивов

void Merge(int \*A, int first, int last) {

int middle, start, final, j;

int \*mas = new int[100];

middle = (first + last) / 2; //вычисление среднего элемента

start = first; //начало левой части

final = middle + 1; //начало правой части

for (j = first; j <= last; j++) //выполнять от начала до конца

if ((start <= middle) && ((final > last) || (A[start] < A[final]))) {

mas[j] = A[start];

start++;

} else {

mas[j] = A[final];

final++;

}

//возвращение результата в список

for (j = first; j <= last; j++)

A[j] = mas[j];

delete[] mas;

};

//рекурсия

void MergeSort(int\* A, int first, int last) {

if (first < last) {

MergeSort(A, first, (first + last) / 2); //сортировка левой части

MergeSort(A, (first + last) / 2 + 1, last); //сортировка правой части

Merge(A, first, last); //слияние двух частей

}

}

int Search\_Binary(int\* arr, int start, int end, int data)

{

int midd=0;

while (1)

{

midd = (start + end) / 2;

if (data < arr[midd]) end = midd - 1;// смещаем правую границу поиска

else if (data > arr[midd]) start = midd + 1;// смещаем левую границу поиска

else { std::cout << "There IS such key in array,and his index is : "; return midd; }

if (start > end)

{

std::cout << "There ISN`T such key in array!"; return -1;

}

}

}

SOURCE.CPP:

#include<iostream>

#include"Header.h"

//Вариант Сортування злиттям.

int main()

{

int i, N = 0, M = 0;

while (N <= 1 || N >= 256 || M <= 1 || M >= 256)

{

std::cout << "Input 1 < N(Array size) < 256 : ";

std::cin >> N;

std::cout << "Input 1 < M(We will be looking for them in array) < 256:";

std::cin >> M;

}

int SIZE = 100;

int\* arr = new int[SIZE];

for (i = 1; i <= N; i++) {

std::cout << i << " element > ";

std::cin >> arr[i];

}

MergeSort(arr, 1, N);

//програма виводить відсортовану послідовність на екран

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

for (i = 1; i <= N; i++)

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

/\* виконує бінарний пошук кожного елементу другої послідовністі x :\*/

int key;

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

{

std::cout << "\n\nEnter your num to search :\n";

std::cin >> key;

std::cout << Search\_Binary(arr, 0, N, key);

}

delete[] arr;

}

Висновок:

виконавши лабораторну роботу №4, я познайомився з алгоритмами сортування та бінарним пошуком. та отримав навички програмування цих алгоритмів.