Министерство образования Республики Беларусь

Учреждение образования

Брестский государственный технический университет

Кафедра ИИТ

Отчет №2

По лабораторной работе

Тема:” ***Бинарный и интерполяционный поиск***”

Выполнил:

Студ. гр.ИИ-23

Романюк А.П.

Проверила:

Глущенко Т. А.

Брест 2023

Вариант №8

***Задание.***

1. Реализовать программно *бинарный* и *интерполяционный* поиск в отсортированном массиве из *10* элементов. Предусмотреть счетчик итераций для нахождения заданного ключа и сравнить значение счетчиков для обоих поисков. Количество тестов – *3*. Последний тест, массив *S = {2,5,8,11,14,17,20,23,25,29}.* Составить сравнительную таблицу по количеству итераций.

Код программы:

#include <iostream>

using namespace std;

int binsearch(int\* a, int n, int target) {

int middle, found;

int iter = 0;

int left = 0, right = n - 1;

if (a[left] > target) {

cout << "bin iter =" << iter << endl;

return 0;

}

if (a[right] < target) {

cout << "bin iter =" << iter << endl;

return right + 1;

}

while (true) {

iter++;

middle = (left + right) / 2;

found = a[middle];

if (found == target) {

cout << "bin iter =" << iter << endl;

return middle;

}

if (middle == left) {

if (a[right] == target) {

cout << "bin iter =" << iter << endl;

return right;

}

cout << "bin iter =" << iter << endl;

return middle + 1;

}

if (found < target) {

left = middle;

}

else {

right = middle;

}

}

}

int interpol(int arr[], int n, int x) {

int lo = 0, hi = (n - 1);

int iter = 0;

while (lo <= hi && x >= arr[lo] && x <= arr[hi]) {

iter++;

if (lo == hi) {

if (arr[lo] == x) return lo;

return -1;

}

int pos = lo + (((double)(hi - lo) / (arr[hi] - arr[lo])) \* (x - arr[lo]));

if (arr[pos] == x) {

cout << "interpolation iter =" << iter << endl;

return pos;

}

if (arr[pos] > x)

hi = pos - 1;

else

lo = pos + 1;

}

return -1;

}

void merge(int arr[], int l, int m, int r) {

int i, j, k;

int n1 = m - l + 1;

int n2 = r - m;

int\* L = new int[n1];

int\* R = new int[n2];

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

L[i] = arr[l + i];

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

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

i = 0;

j = 0;

k = l;

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(int arr[], int l, int r) {

if (l < r) {

int m = l + (r - l) / 2;

mergeSort(arr, l, m);

mergeSort(arr, m + 1, r);

merge(arr, l, m, r);

}

}

int main() {

int n, target;

cout << "inter number of elements\n";

cin >> n;

int\* a = new int[n];

cout << "enter numbers\n";

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

cin >> a[i];

}

cout << "enter target element\n";

cin >> target;

int choise;

cout << "enter choise\n";

cin >> choise;

switch (choise)

{

case 1:

cout << binsearch(a, n, target);

break;

case 2:

cout << interpol(a, n, target);

break;

case 3:

cout << binsearch(a, n, target)<<endl;

cout << interpol(a, n, target)<<endl;

break;

case 4:

mergeSort(a, 0, n-1);

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

cout << a[i] << " ";

}

break;

}

}



