МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ

Кафедра системного проектування

**ЗВІТ**про виконання лабораторної роботи №3

«Дослідження елементарних алгоритмів сортування та пошуку»

з дисципліни «Алгоритми та структури даних»

Виконав:

cтудент І курсу

групи ДА-02

Гринчишин Д.Б.

Перевірили:

..

Київ – 2020

**Мета**. Ознайомитись і дослідити елементарні алгоритми сортування: “бульбашкою”, вибором, вставками; алгоритм перетасовки Фішера-Йєтса та алгоритми пошуку: лінійний пошук, бінарний пошук. Набути навичок їх реалізації мовою програмування С/C++ та порівняти ці алгоритми.

**Варіант завдань - 2**

**Завдання 1.**

Базове сортування “бульбашкою”, пошук елементів у масиві

1.1 Створити масив на N = 1000 елементів та заповнити його випадковими цілими числами від -2000 до 2000

1.2 Написати функцію для сортування алгоритмом “бульбашкою”, відсортувати масив за зростанням 1.3 Написати функцію для пошуку елементу в масиві лінійним алгоритмом, яка повертає індекс знайденого елементу або -1, якщо такий елемент не знайдено

1.4 Написати функцію для пошуку елементу в масиві бінарним алгоритмом, яка повертає індекс знайденого елементу або -1, якщо такий елемент не знайдено

1.5 Порівняти швидкодію алгоритмів пошуку (пошук у масиві випадкових чисел від -2000 до 2000)

**Код:**

#include<time.h>

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

#include<stdbool.h>

#include<time.h>

const int N = 1000;

int temp;

int \*bubble(int mass[])

{

for (int iter\_glob = 1; iter\_glob < N; iter\_glob++)

{

for (int iter\_loc = 0; iter\_loc < N - iter\_glob; iter\_loc++)

{

if (mass[iter\_loc] > mass[iter\_loc + 1])

{

temp = mass[iter\_loc + 1];

mass[iter\_loc + 1] = mass[iter\_loc];

mass[iter\_loc] = temp;

}

}

}

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

{

printf("%d\n", mass[i]);

}

return mass;

}

int linear(int elem, int mass[])

{

int out = -1;

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

{

if (mass[i] == elem) return i;

}

return out;

}

int bin(int elem, int mass[])

{

float step = (float)N / 4;

float middle = (float)N / 2;

while (elem != mass[(int)middle])

{

if (step < 0.5) return -1;

middle += step \* (elem > mass[(int)middle] ? 1 : -1);

//printf("%f %f\n", middle, step);

step /= 2;

}

return middle;

}

int main()

{

int index;

int mass[N];

int \*sorted;

clock\_t start;

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

{

mass[i] = rand() % 4000 - 2000;

//printf("%d\n", mass[i]);

}

sorted = bubble(mass);

start = clock();

for (int i = 0; i < 100000; i++) { index = linear(sorted[450], sorted); }

printf("\n%d time: %f", index, float(clock() - start) / CLOCKS\_PER\_SEC);

start = clock();

for (int i = 0; i < 100000; i++) { index = bin(sorted[450], sorted); }

printf("\n%d time: %f", index, float(clock() - start) / CLOCKS\_PER\_SEC);

return 1;

}



**Завдання 2.**

Сортування елементарними алгоритмами по різним критеріям

2.1 Створити масив на N = 200 елементів та заповнити його послідовними числами від 1 до N

2.2 Написати функцію для перетасовки масиву алгоритмом ФішераЙєтса, перетасувати масив

2.3 Написати функцію для сортування алгоритмом вибором / вставками, відсортувати масив за певним критерієм (варіант 2)

**Код:**

#include<time.h>

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

#include<stdbool.h>

#include<time.h>

const int N = 200;

int temp, r;

clock\_t start;

int \*permutat(int mass[])

{

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

{

r = rand() % i;

temp = mass[i];

mass[i] = mass[r];

mass[r] = temp;

}

return mass;

}

int max(int mass[])

{

int n = mass[0], len = sizeof(mass);

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

{

if (n < mass[i])

{

n = mass[i];

}

}

return n;

}

int \*sort(int mass[])

{

int n1, n2;

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

{

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

{

n1 = mass[i], n2 = mass[j];

int arr1[3] = { n1 % 10, n1 / 10 % 10, n1 / 100 };

int arr2[3] = { n2 % 10, n2 / 10 % 10, n2 / 100 };

int max1 = max(arr1);

int max2 = max(arr2);

if (max1 > max2)

{

mass[j] = n1;

mass[i] = n2;

}

else if(max1 == max2)

{

if (n1 >= n2)

{

mass[j] = n1;

mass[i] = n2;

}

}

}

}

return mass;

}

int main()

{

int mass[N];

int \*permutated;

int \*sorted;

for (int i = 0; i < N; i++) { mass[i] = i + 1; }

permutated = permutat(mass);

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

{

printf("%d ", permutated[i]);

}

printf("\n\n");

sorted = sort(mass);

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

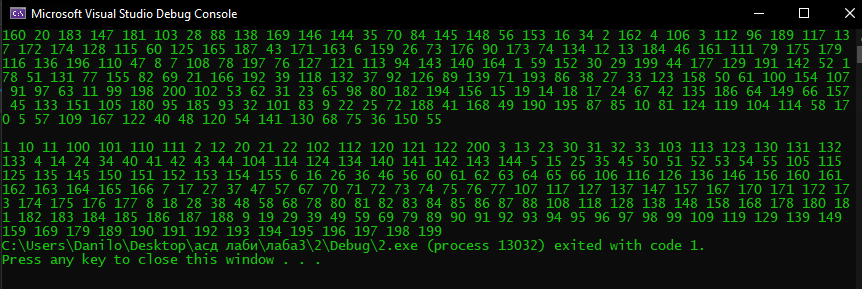
{

printf("%d ", sorted[i]);

}

return 1;

}



**Висновок:**

Я здобув конкретні навички та знання в написанні коду на мові програмування C, а саме набув навичок в реалізації таких елементарних алгоритмів сортування та пошуку, як: сортування “бульбашкою”, вибором, вставками; алгоритм перетасовки Фішера-Йєтса та алгоритми пошуку: лінійний пошук, бінарний пошук. Навчився заміряти час виконання алгоритмів та порівнювати їх складність.

**Контрольні запитання:**

1) Алгоритм сортування бульбашкою порівнює усі елементи масиву, не здійснюючи перестановок, якщо масив уже відсортований. Можна оптимізувати таким чином : у випадку, якщо не було здійснено жодної перестановки упродовж першої ітерації, зупинити алгоритм.

2) Олінійного = n

Обінарного = log2(n)

Лінійний пошук добре справляється з як не відсортованим масивом так із відсортованим, а бінарний пошук працює значно швидше на відсортованому масиві, але не працює взагалі на не відсортованому масиві.

3)Сортування вибором відбувається за допомогою перестановок всередині масиву, а сортування вставками відбувається за допомогою створення 2 масивів.

.