Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра системотехніки

Дисципліна: «Теорія Алгоритмів»

**ЛАБОРАТОРНА РОБОТА № 5**

**«АЛГОРИТМИ ПОШУКУ ТА СОРТУВАННЯ»**

|  |  |  |
| --- | --- | --- |
| Виконав:  ст. гр. ІТУ-19-2  Куренков Богдан Михайлович |  | Прийняв:  доцент  Ситнікова Поліна Едуардівна  з оцінкою «\_\_\_\_\_\_\_\_\_\_\_\_»  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_р. |

Харків 2019

**Мета роботи**

**Мета роботи:** Ознайомлення на практичних прикладах з основними алгоритмами пошуку та сортування масивів. Визначення складності та порівняння існуючих алгоритмів.

**Завдання**

Розробити алгоритм сортування масиву з 40 цілих чисел (або матриці 4на10) у відповідності зі своїм варіантом. Оцінити складність алгоритму. Розробити програму на С++, що реалізує заданий алгоритм. Протестувати алгоритм на різних масивах.

**Завдання 1**

**Варіант 10.** Відсортувати елементи на парних місцях за спаданням, а на непарних – за зростанням. Використовувати метод прямого вибору. Зробити лічильник операцій обміну та лічильник операцій порівняння. Оцінити складність алгоритму.

**Код програми**

#include <iostream>

#include <ctime>

using namespace std;

int main() {

srand(time(0));

const int size = 40;

int arr[size], min=0, max=0, compare\_counter = 0, swap\_counter = 0, tmp\_index;

cout << "Unsorted massive:" << endl;

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

arr[i] = rand() % 21 - 10;

cout.width(4);

cout << arr[i];

}

cout << endl;

for (int j = 1; j < size-2; j+=2) {

min = arr[j];

for (int i = j+2; i < size; i+=2) {

compare\_counter++;

if (arr[i] < min) {

min = arr[i];

tmp\_index = i;

}

}

swap\_counter++;

arr[tmp\_index] = arr[j];

arr[j] = min;

}

for (int j = 0; j < size-2; j += 2) {

max = arr[j];

for (int i = j+2; i < size; i += 2) {

compare\_counter++;

if (arr[i] > max) {

max = arr[i];

tmp\_index = i;

}

}

swap\_counter++;

arr[tmp\_index] = arr[j];

arr[j] = max;

}

cout << "Sorted massive:" << endl;

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

cout.width(4);

cout << arr[i];

}

cout <<endl<< "Pairs:" << endl;

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

if (i % 2 == 0) {

cout.width(4);

cout << arr[i];

}

}

cout <<endl<< "Unpairs:"<< endl;

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

if (i % 2 == 1) {

cout.width(4);

cout << arr[i];

}

}

cout << "\n\nNumber of compares:" << compare\_counter;

cout << "\nNumber of swaps:" << swap\_counter << "\n\n";

system("pause");

return 0;

}

**Складність алгоритму:**

**Завдання 2**

**Варіант 2**. Реалізувати алгоритм пошуку в динамічному масиві заданої послідовності елементів (пошук образу в рядку). Використовувати метод прямого пошуку. Встановити лічильник операцій порівняння. Зробити висновки щодо їх кількості.

**Код програми**

#include <iostream>

using namespace std;

int main() {

int size, t = 0, j = 0, p = 0, compare\_counter = 0;

int\* arr\_control = new int[10];

cout << "Started massive: ";

for (int i = 0; i < 10; i++)

arr\_control[i] = i, cout << arr\_control[i] << " ";

cout << "\nEnter the required spirits massive size: ";

cin >> size;

cout << "Fill it!: ";

int\* arr\_dynamical = new int[size];

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

cin >> arr\_dynamical[i];

cout << "Spirits massive: ";

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

cout << arr\_dynamical[i] << " ";

cout << "\n\n";

for (int i = 0; i < 10 - size + 1; i++) {

compare\_counter++;

if (arr\_control[i] == arr\_dynamical[j]) {

t = i;

t++;

j++;

if (j == size) {

p++;

j = 0;

}

}

while (arr\_control[t] == arr\_dynamical[j]) {

compare\_counter++;

t++;

j++;

if (j == size) {

p++;

j = 0;

break;

}

}

j = 0;

}

cout << "Count of spirits:" << p << endl;

cout << "Count of comparisons:" << compare\_counter << "\n\n";

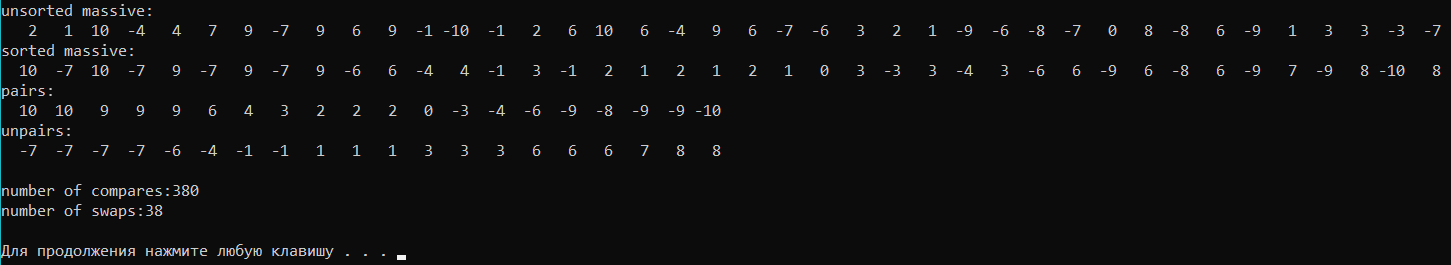
system("pause");

return 0;

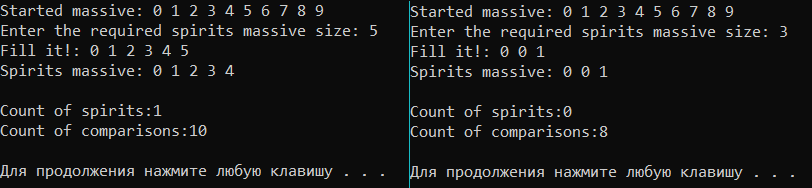
}

**Складність алгоритму:**

**Результат роботи до завдання 1**

**

**Результат роботи до завдання 2**

**

**Висновки.** На практичних прикладах ми ознайомилися з основними алгоритмами пошуку та сортування масивів та методами оцінки алгоритмів.