|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ** |
| **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** |
| **«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»** |
|  |
| Институт информационных технологий и управления в технических системах |
| (полное название института) |
|  |
| кафедра «Информационные системы» |
| (полное название кафедры) |

**ОТЧЁТ**

по лабораторной работе №6

на тему

«АНАЛИЗ АЛГОРИТМОВ СОРТИРОВКИ»

Выполнил

студент ИИТУТС

группы ИС/б-18-2-о

Радыгина Екатерина

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | | |
|  | (должность, учёная степень преподавателя) | | | | | | | | |
|  |  | | | | | | | | |
|  | (ФИО преподавателя) | | | | | | | | |
|  | « |  | » |  |  |  | 20 | 20 | г. |
|  |  | | | | | | | | |
|  | (оценка) | | | | | | | | |
|  |  | | | | | | | | |
|  |  | | | | | | | | |

Севастополь 2020

**6.1 Цель работы**

Научиться оценивать сложность и количество операций для алгоритмов сортировки.

**6.2 Вариант задания**

Вариант 16

Для каждого из приведённых алгоритмов найдите оценку для количества шагов.

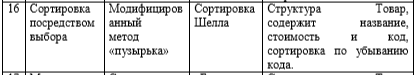


Рисунок 6.1 – Вариант задания

**6.3 Ход работы**

6.3.1 Напишем программу реализующую разные типы сортировок.

Текст программы

#include <iostream>

#include <string>

using namespace std;

const int n = 10;

struct tovar {

char\* name[n] = {"a","ma","klass","guchi","flip","flap","boop","door","soojin","leto"};

float price[n] = {1234,134,1,4232,545234,423,77,12,875,12356};

int id[n] = {3,6,1,7,9,8,2,4,5,10};

};

//Сортировка посредством выбора

void VuborSort(int a[], int n){

int counter = 0;

int reassign = 0;

for (int i=n, imax; i>1; i--){

imax = 1;

counter++;

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

counter++;

if (a[j] < a[imax]){

imax = j;

reassign++;

}

}

swap(a[i],a[imax]);

}

cout << "Количество итераций: " << counter << endl;

cout << "Количество переприсваиваний: " << reassign << endl;

}

//Сортировка массива пузырьком

void bubbleSort(int a[], int n){

int counter = 0;

int reassign = 0;

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

counter++;

for (int j=i+1; j<n; j++){

counter++;

if (a[i] < a[j]){

reassign++;

swap(a[i],a[j]);

}

}

}

cout << "Количество итераций: " << counter << endl;

cout << "Количество переприсваиваний: " << reassign << endl;

}

//Сортировка Шелла

void shellSort(int a[], int n) {

int i, j, temp, step = n/2;

int counter = 0;

int reassign = 0;

while (step > 0) {

counter++;

for (i=0; i<(n-step); i++){

j = i;

counter++;

while (j>=0 && a[j] > a[j+step]) {

reassign++;

swap(a[j],a[j+step]);

j--;

}

}

step = step/2;

}

cout << "Количество итераций: " << counter << endl;

cout << "Количество переприсваиваний: " << reassign << endl;

}

int main() {

setlocale(LC\_ALL, "Russian");

tovar a;

cout << "Сортировка выбором\n";

VuborSort(a.id, n);

cout << endl << "Сортировка пузырьком\n";

bubbleSort(a.id, n);

cout << endl << "Сортировка Шелла\n";

shellSort(a.id, n);

}

На рисунке 6.2 представлен пример выполнения программы.

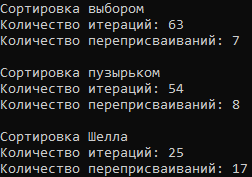


Рисунок 6.2 – Выполнение программы

**Вывод**

В ходе лабораторной работы мы научились оценивать сложность и количество операций для алгоритмов сортировки. Самой быстрой из предложенных алгоритмов сортировок оказалась сортировка Шелла, после неё – сортировка выбором, и самой медленной – сортировка пузырьком.