МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1**

**по дисциплине  
 «АЛГОРИТМЫ И АНАЛИЗ СЛОЖНОСТИ»**

Выполнил студент группы 24/2                                       А.А.Иванов

Направление подготовки  02.03.03  Математическое обеспечение и администрирование информационных систем

Курс    2

Отчет принял доктор физико-математических наук, профессор                                                                                       А.И. Миков

Краснодар

2022 г.

**Задание**: Дан произвольный массив, который необходимо отсортировать. Найти зависимость сложности алгоритма от количества чисел в массиве.

**Решение**.

Массив будем сортировать пузырьковой сортировкой, так как она является самой простой. Будем рассматривать 3 случая: лёгкое заполнение (массив уже отсортирован), сложное заполнение (массив расположен по убыванию), случайное заполнение (массив заполняется числами в случайной последовательности).

**Текст программы:**

#include <iostream>

#include <time.h>

#include <stdlib.h>

using namespace std;

int main ()

{

setlocale (LC\_ALL, "rus");

srand(time(NULL));

//int n;

int kolopobsh = 0;

int srzn = 0;

cout << "Введите колличество элементов массива:" << endl;

//cin >> n;

//for (int i1 = 0; i1 < 1000; i1++) //повторение 1000 раз

//{

for (int n = 10; n < 101; n += 10)

{

int kolop = 0;

int\* arr = new int[n];

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

{

//arr[i] = rand() % 100 + 1; //случайное заполнение

//arr[i] = i; //лёгкое заполнение

//arr[i] = n - i; //сложное заполнение

}

cout << "Ваш массив:" << endl;

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

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

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

{

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

{

if (arr[j] > arr[j + 1])

{

int a = arr[j];

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

arr[j + 1] = a;

kolop += 5;

}

kolop += 2;

}

}

cout << endl;

cout << "Отсортированный массив:" << endl;

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

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

cout << endl;

cout << "Колличество операций - " << kolop << endl;

cout << endl;

kolopobsh += kolop;

}

//}

srzn = kolopobsh / 1000;

cout << "Среднее колличество операций для 1000 запусков - " << srzn << endl;

system("pause");

}

**После работы программы для 3-ёх случаев получаем следующие графики и таблицы:**

|  |  |
| --- | --- |
| Случайное заполнение | |
| Кол-во элементов | Сред. кол-во опер-ий |
| 10 | 292 |
| 20 | 1231 |
| 30 | 2815 |
| 40 | 5041 |
| 50 | 7933 |
| 60 | 11435 |
| 70 | 15643 |
| 80 | 20431 |
| 90 | 25900 |
| 100 | 32052 |
| Для каждого значения - 1000 запусков | |
| Диапазон значений элементов - от 1 до 100 | |

|  |  |
| --- | --- |
| Самое лёгкое заполнение | |
| Кол-во элементов | Сред. кол-во опер-ий |
| 10 | 180 |
| 20 | 760 |
| 30 | 1740 |
| 40 | 3120 |
| 50 | 4900 |
| 60 | 7080 |
| 70 | 9660 |
| 80 | 12640 |
| 90 | 16020 |
| 100 | 19800 |
| Для каждого значения - 1 запуск | |
| Диапазон значений элементов - от 1 до 100 | |

|  |  |
| --- | --- |
| Самое сложное заполнение | |
| Кол-во элементов | Сред. кол-во опер-ий |
| 10 | 405 |
| 20 | 1710 |
| 30 | 3915 |
| 40 | 7020 |
| 50 | 11025 |
| 60 | 15930 |
| 70 | 21735 |
| 80 | 28440 |
| 90 | 36045 |
| 100 | 44550 |
| Для каждого значения - 1 запуск | |
| Диапазон значений элементов - от 1 до 100 | |

**Теперь вычислим формулу для нашего алгоритма:**

Будем пользоваться методом неопределённых коэффицентов и получим следующую систему:

, где *n* > 0 (общий вид формулы)

Случайное заполнение

Лёгкое заполнение

Сложное заполнение

Коэф-ты для случайного заполнения: а = ; в = - ; с = ;

Коэф-ты для лёгкого заполнения: а = 2; в = -2; с = 0;

Коэф-ты для сложного заполнения: а = 4,5; в = -4,5; с = 0;

**Вывод**: средняя сложность представленного алгоритма пузырьковой сортировки массива размерности n составляет:

, где *n* > 0.