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 № 3**

**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:** найти зависимость временной сложности рекурсивного алгоритма Хоара быстрой сортировки массива чисел от количества чисел в массиве, , при n от 100 до 1000 с шагом 100.

**Решение:** напишем алгоритм рекурсивной быстрой сортировки Хоара, посчитаем количество операций, и определим временную сложность.

**Код программы:**

#include <iostream>

#include <ctime>

#include <cstdlib>

#include <random>

using namespace std;

int kolop = 0;

void HoarFastSort(int\* arr, int first, int last)

{

int i = first;

int j = last;

double tmp, x = arr[(first + last) / 2];

kolop += 3;

do

{

while (arr[i] < x)

{

i++;

kolop++;

}

while (arr[j] > x)

{

j--;

kolop++;

}

if (i <= j)

{

if (i < j)

{

tmp = arr[i];

arr[i] = arr[j];

arr[j] = tmp;

kolop += 3;

}

i++;

j--;

kolop += 2;

}

kolop++;

} while (i <= j);

if (i < last)

HoarFastSort(arr, i, last);

if (first < j)

HoarFastSort(arr, first, j);

}

int main()

{

setlocale(LC\_ALL, "rus");

srand(time(NULL));

for (int n = 100; n < 1001; n += 100)

{

for (int n1 = 0; n1 < 1000; n1++)

{

int\* arr = new int[n];

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

arr[i] = rand() % 101;

//cout << "Ваш массив:" << endl;

//cout << endl;

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

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

HoarFastSort(arr, 0, n - 1);

//cout << endl << endl;

//cout << "Ваш отсортированный массив:" << endl << endl;

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

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

//cout << endl << endl;

//cout << "Колличество операций при сортировке: " << kolop << "." << endl;

//cout << endl;

}

cout << "Среднее колличество операций после 1000 запусков (" << n << "): " << kolop / 1000 << "." << endl;

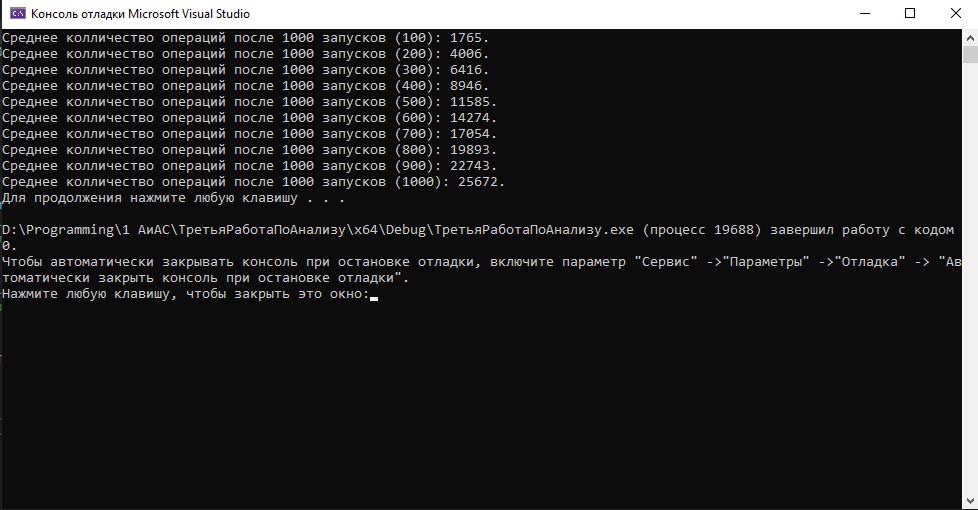
kolop = 0;

}

return 0;

}

**Пример вывода в консоль:**

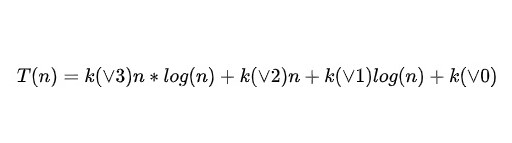


**Таблица значений:**

|  |  |
| --- | --- |
| Кол-во элементов в массиве | Среднее кол-во операций |
| 100 | 1767 |
| 200 | 3993 |
| 300 | 6419 |
| 400 | 8957 |
| 500 | 11585 |
| 600 | 14282 |
| 700 | 17074 |
| 800 | 19855 |
| 900 | 22760 |
| 1000 | 25653 |

**График:**

**Исходная формула:**



**Составим систему из 4 ур-ий:**

k3n1\*logn1+k2n1+k1logn1+k0=y1=T(n1)

k3n2\*logn2+k2n2+k1logn2+k0=y2=T(n2)

k3n3\*logn3+k2n3+k1logn3+k0=y3=T(n3)

k3n4\*logn4+k2n4+k1logn4+k0=y4=T(n4)

**Подставим наши значения** (за n - 100, 300, 600, 900, за y - 1767, 6419, 14282, 22760), **и решим систему методом Гаусса:**

k3n1\*logn1+k2n1+k1logn1+k0=y1=T(n1)

k3n2\*logn2+k2n2+k1logn2+k0=y2=T(n2)

k3n3\*logn3+k2n3+k1logn3+k0=y3=T(n3)

k3n4\*logn4+k2n4+k1logn4+k0=y4=T(n4)

**Наши коэффиценты:** k3 = 2.64 , k2 = -0.88, k1 = 46.87 , k0 = -197.12 .

**Подставим коэффиценты в исходную формулу:**

T(n) = 2.64\*n\*log(n)+(-0.88)n+46.87\*log(n)+(-197.12)