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 25/2                                       А.А. Козин

Направление подготовки  02.03.03  Математическое обеспечение и администрирование информационных систем

Курс    2

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

Краснодар

2021 г.

**Задание**: Найти функцию сложности для алгоритма умножения квадратных матриц.

**Решение**.

Для анализа сложности алгоритма написана программа, которая считает и выдает количество простых операций при умножении двух матриц. Длина массивов меняется от 2 до 30.

**Код программы:**

#include <iostream>

#include <ctime>

using namespace std;

void main()

{

setlocale(LC\_ALL, "RU");

srand(time(0));

int counter = 0;

cout << "n" << " " << "counter" << " " << "T(n)" << endl;

cout << "-----------------------------------------------" << endl;

for (int n = 2; n <= 30; n++)

{

int\*\* FirstArray; FirstArray = new int\* [n];

int\*\* SecondArray; SecondArray = new int\* [n];

int\*\* ThirdArray; ThirdArray = new int\* [n];

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

{

FirstArray[i] = new int[n];

SecondArray[i] = new int[n];

ThirdArray[i] = new int[n];

}

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

{

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

{

FirstArray[i][j] = rand() % 15 + 3;

SecondArray[i][j] = rand() % 15 + 3;

}

}

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

{

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

{

ThirdArray[i][j] = 0;

counter++;

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

{

ThirdArray[i][j] += FirstArray[i][k] \* SecondArray[k][j];

counter += 3;

}

}

}

cout << n << " ";

cout << counter << " ";

cout << " T(" << n << ")=" << "k(3)\*" << n \* n \* n << "\*k(2)\*" << n \* n << "\*k(1)" << endl;

counter = 0;

}

system("pause>nul");

}

Программа выдает результат в консоль в следующем виде:

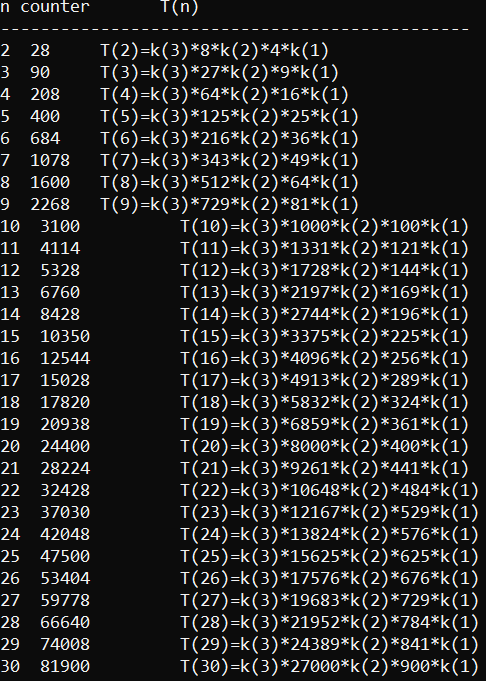


График роста среднего количества операций для сложности (объема) исходных данных *n* в диапазоне от 2 до 30 выглядит следующим образом:

**Анализ зависимости:**

Решаем задачу методом неопределенных коэффициентов. Полученный в результате расчетов график очень близок к графику зависимости:

, где *n* > 0

Найдем неизвестные коэффициенты

Возьмем любые три точки на графике (*n* = 5, *y* = 400); (*n* = 15, *y* = 10350); (*n* = 30; *y* = 81900) и решим систему из 3 уравнений:

;;

Получим формулу вида:

где *n* > 0.

**Вывод**: Сложность умножения матриц из *n* элементов имеет вид

, где *n* > 0.