|  |  |
| --- | --- |
| для прик эмбл | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования"Российский технологический университет"МИРЭА | |
| **Факультет информационных технологий** | |
| **Кафедра КБ-3 «Управление и моделирование систем»** | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНЫМ РАБОТАМ** | |
| **по дисциплине** | |
| 1. **«Анализ сложности алгоритмов»** | |
|  | |
| Выполнил студент группы БСБО-02-18 | *Каталев П.А.* |
| Принял  Доцент, заместитель заведующего кафедры | *Филатов В.В.* |

|  |  |  |
| --- | --- | --- |
| Лабораторные работы выполнены | «\_\_»\_\_\_\_\_\_\_2019 г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2019 г. |  |

Москва 2019

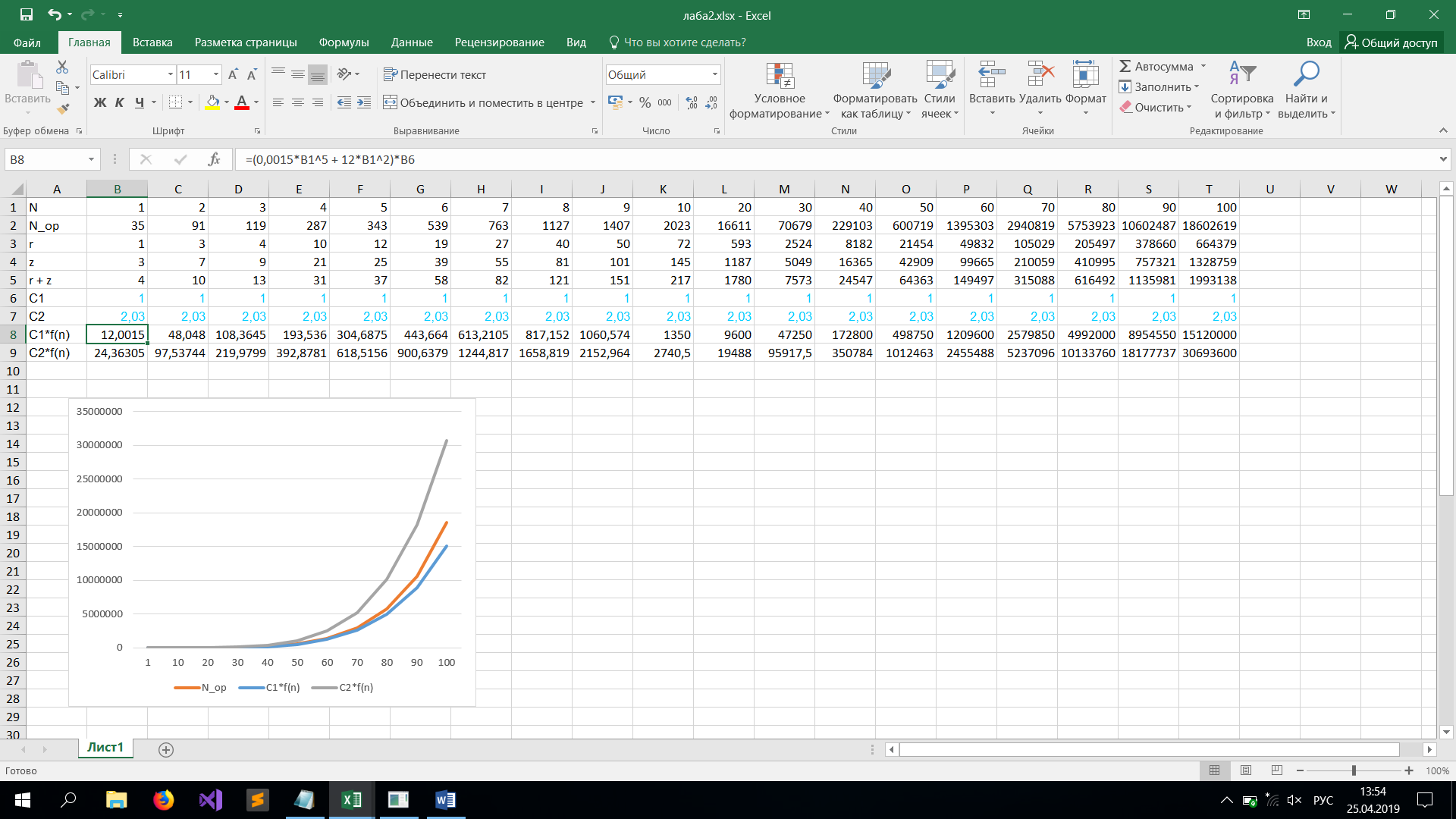
***Вариант №314***

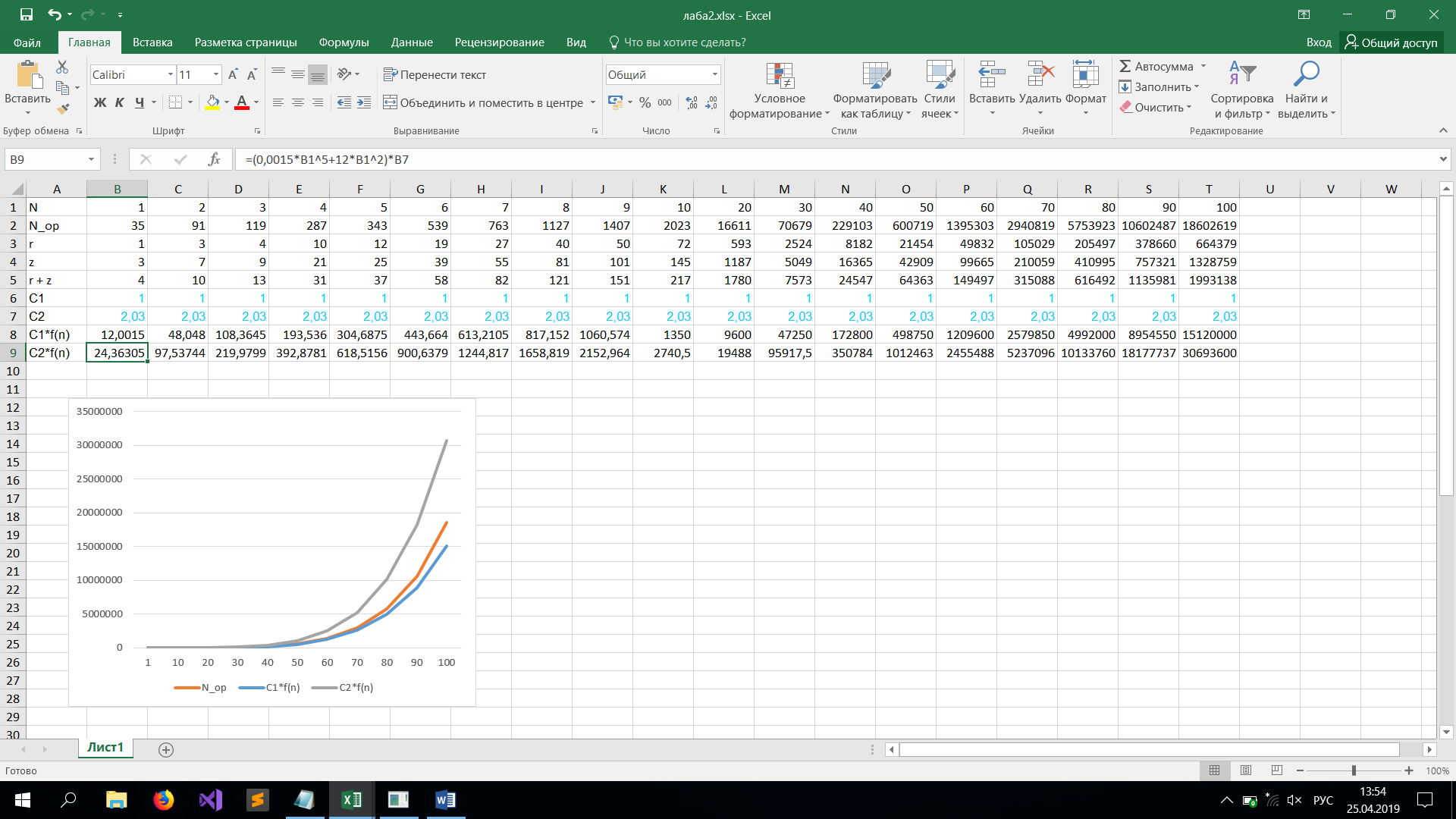
***Задание:***

• Построить дерево рекурсии (Минимум 20 узлов);   
• Построить таблицу до такого значения N, при котором программа будет работать приблизительно 5 минут.;   
N |1 | 2 | … | 10 | 20 | 40 | 60 | 80 | 100   
N\_op   
r   
r + z   
Вид   
f(n)   
C1(f(n))   
C2(f(n))   
  
• Заполнить таблицу данными из компиляций программ.;   
• Построить графики функций N\_op, C1(f(n)), C2(f(n));   
• Подобрать примерный полином исходя из графика N\_op и записать его в таблицу.

***Полученные результаты работы:***

Предполагаемый вид полинома функции роста *f(n)*, и соответствующие ей *Θ(f(n))*:





***Таблица:***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | N\_op | r | z | r + z | C1 | C2 | C1\*f(n) | C2\*f(n) |
| 1 | 35 | 1 | 3 | 4 | 1 | 2,03 | 12,0015 | 24,36305 |
| 2 | 91 | 3 | 7 | 10 | 1 | 2,03 | 48,048 | 97,53744 |
| 3 | 119 | 4 | 9 | 13 | 1 | 2,03 | 108,3645 | 219,9799 |
| 4 | 287 | 10 | 21 | 31 | 1 | 2,03 | 193,536 | 392,8781 |
| 5 | 343 | 12 | 25 | 37 | 1 | 2,03 | 304,6875 | 618,5156 |
| 6 | 539 | 19 | 39 | 58 | 1 | 2,03 | 443,664 | 900,6379 |
| 7 | 763 | 27 | 55 | 82 | 1 | 2,03 | 613,2105 | 1244,817 |
| 8 | 1127 | 40 | 81 | 121 | 1 | 2,03 | 817,152 | 1658,819 |
| 9 | 1407 | 50 | 101 | 151 | 1 | 2,03 | 1060,574 | 2152,964 |
| 10 | 2023 | 72 | 145 | 217 | 1 | 2,03 | 1350 | 2740,5 |
| 20 | 16611 | 593 | 1187 | 1780 | 1 | 2,03 | 9600 | 19488 |
| 30 | 70679 | 2524 | 5049 | 7573 | 1 | 2,03 | 47250 | 95917,5 |
| 40 | 229103 | 8182 | 16365 | 24547 | 1 | 2,03 | 172800 | 350784 |
| 50 | 600719 | 21454 | 42909 | 64363 | 1 | 2,03 | 498750 | 1012463 |
| 60 | 1395303 | 49832 | 99665 | 149497 | 1 | 2,03 | 1209600 | 2455488 |
| 70 | 2940819 | 105029 | 210059 | 315088 | 1 | 2,03 | 2579850 | 5237096 |
| 80 | 5753923 | 205497 | 410995 | 616492 | 1 | 2,03 | 4992000 | 10133760 |
| 90 | 10602487 | 378660 | 757321 | 1135981 | 1 | 2,03 | 8954550 | 18177737 |
| 100 | 18602619 | 664379 | 1328759 | 1993138 | 1 | 2,03 | 15120000 | 30693600 |

***Программа:***

#include "iostream"

#define VAR 314 // Указать три последние цифры своего студ шифра

int y, n, r, N\_op, z;

int f(int x)

{ N\_op++; // ( x )

n++; N\_op++; // ++

std::cout<< " f("<<x<<")"; N\_op++; // cout

N\_op++; // (x>0)

if (x>0)

{

r++; N\_op++; // ++

/\* (\*) \*/

/// return f(x/3)+ f(x-2)+f(0); N\_op+=8; // / (.) + - (.) + (.) return

N\_op+=9;

return f(x / 2) + f(x - 2) \* f(3\*x/5);

}

else

{

std::cout<< std::endl; N\_op++; // cout

z++; N\_op++; N\_op++; // ++

return 1; N\_op++; // return

}

}

void variant(void)

{

int var = (VAR<100)?VAR+100:VAR;

int v1, v2, v3;

v1 = var % 10 ;

v2 = var / 10 % 10;

v3 = var / 100 % 10;

std::cout <<std::endl<<std::endl<< "(Student: "<<VAR<<")----> f";

if (v1 > 5) std::cout <<"(x - "<<v1%3+1<<")";

else std::cout <<"(x / "<<v1%2+2<<")";

std::cout << " + f";

if (v2 < 5) std::cout <<"(x - "<<v2%3+1<<")";

else std::cout <<"(x / "<<v2%2+2<<")";

if (((v3+v2+v1)%3)==0) std::cout << " - f("<<v1+1<<"\*x/"<<v1+3<<")" << std::endl;

if (((v3+v2+v1)%3)==1) std::cout << " + f("<<"x -"<<v2+1<<") " << std::endl;

if (((v3+v2+v1)%3)==2) std::cout << " \* f(3\*x/5)" << std::endl;

}

int main(void)

{

int N = 10;

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

n=0; // Общее количество вызовов

r= 0; // Общее количество внутренних узлов

z= 0; // Общее количество листьев

N\_op=0; // Общее количество инструкций

// Глубина рекурсии

f(i);

std::cout<<" N = "<< i << " r = "<< r << " z = " << z<< " N\_op ="<<N\_op<<std::endl;

}

variant();  
}