†иыгФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

(НИУ «БелГУ»)

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ

КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

**Отчет по лабораторной работе 1**

**по дисциплине: «Алгоритмы и анализ сложности»**

**Тема работы «Оценка временной сложности алгоритмов»**

студента очного отделения

3 курса 12001801 группы

Капустина Виктора Сергеевича

Проверил(а):

Гайворонский Виталий Александрович

Белгород 2021

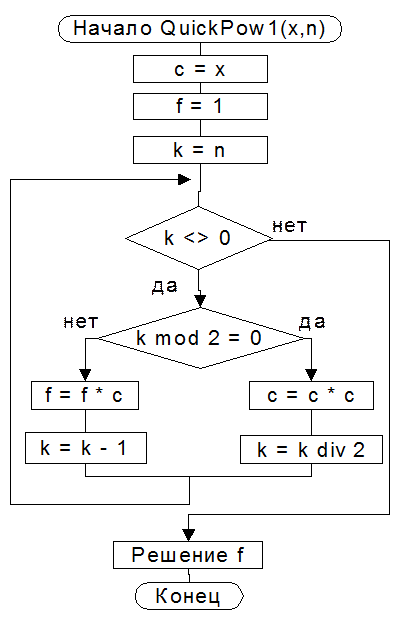
**Цель работы:** Провести анализ и оценку временной сложности заданного алгоритма.

**Ход работы**

Рассмотрим 4 вариант. Требуется провести анализ и оценку временной сложности алгоритма быстрого возведения в степень.

Алгоритм предназначен для решения следующей задачи: дано число *x* и целое неотрицательное число n>=0. Вычислить значение функции 

Его алгоритм приведен на рисунке 1.

  
Рис.1 Алгоритм быстрого возведения в степень

Проведем его анализ. Входными значениями будут являться: Число, возводимое в степень, и сама степень, которая может служить мерой объема входных. Все действия вне цикла, а то есть операторы присваивания имеют некоторое постоянное время выполнения и не зависит от входных данных. Все выполняется последовательно и имеет временную сложность порядка *О(1).*

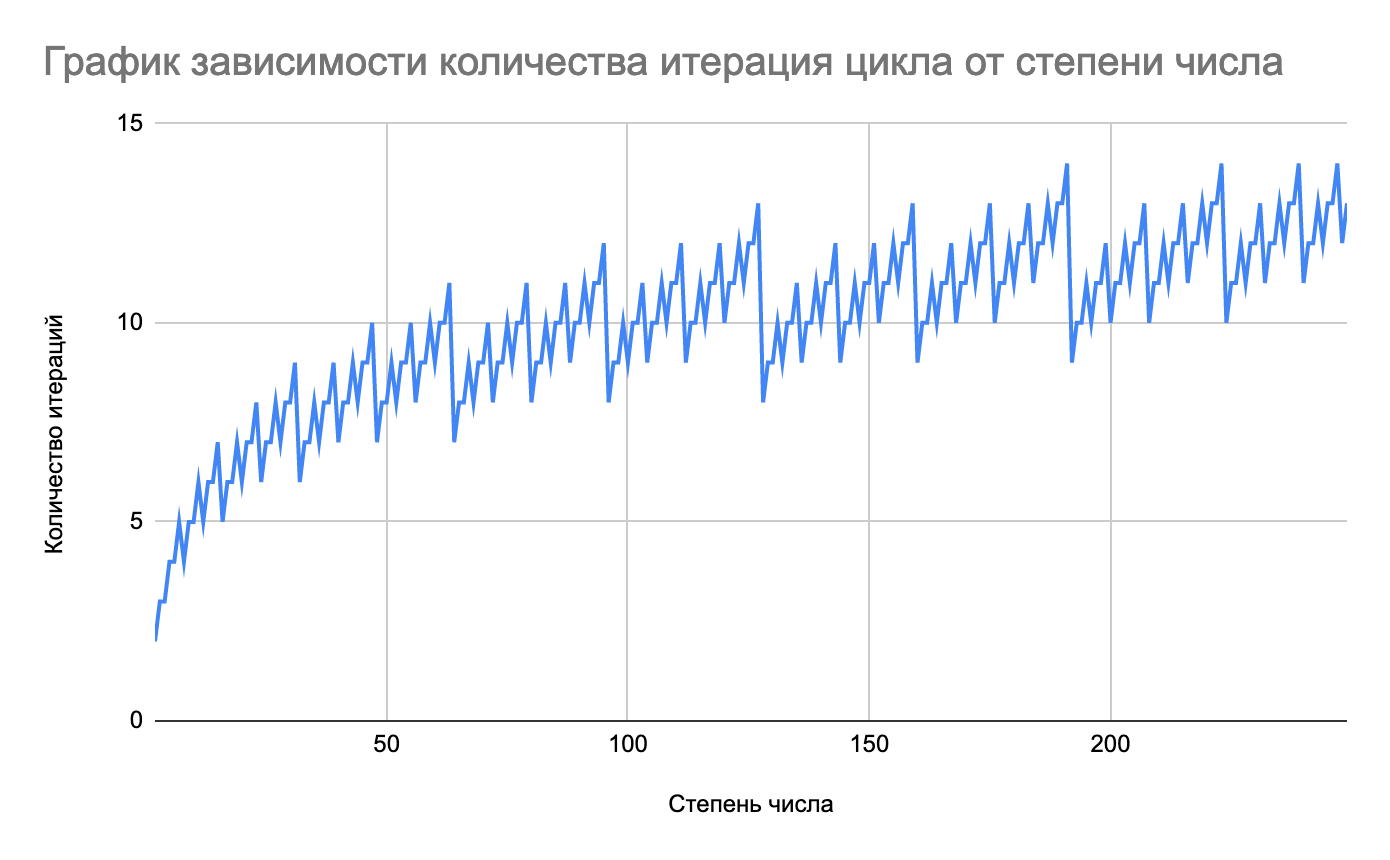
При анализе цикла, можно заметить, что количество его итераций должно зависеть от входного значения степени. Внутренность цикла также имеет временную сложность порядка *О(1)* и сложность всего алгоритма будет зависеть только от количества итераций цикла.

Попробуем определить количество итераций цикла в зависимости от *n* проведя ряд тестов. Таблица 1 содержит динамику изменения количества итераций в зависимости от значения степени.

Таблица 1.

| Степень | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Количество итераций | 1 | 2 | 3 | 3 | 4 | 4 | 5 | 6 | 5 | 5 | 6 | 5 | 6 | 6 | 7 |

Как видно по данным Таблицы 1, количество итераций все же зависит от степени, что является одним из входных в данных алгоритма. Сделав по полученным данным график(Рис. 1) мы можем увидеть, что в общей картине, график имеет логарифмический вид. Из-за того, что все действия выполняются не параллельно, то согласно правилу сумм, временная сложность алгоритма будет равна большему значению из всех. Все преобразования имеют сложность порядка *О(1),* а временная сложность цикла равна *О(log n)*, то и временная сложность всего алгоритма будет *О(log n).* Т.к. только цикл оказывает влияния на временную сложность алгоритма, то и нижняя и верхняя оценка будут равны сложности цикла, т.е. *О(log n)*

  
Рис. 1 График зависимости

**Вывод:** В лабораторной работе проведен анализ алгоритма быстрого возведения в степень. Его временная сложность имеет верхнюю и нижнюю оценку равными *О(log n).*

**Листинг программы**

#include <iostream>

using namespace std;

int main()

{

int f=1;

int n=2;

int st=0;

int counter=0;

for(int i=2;i<16;i++)

{

st=i;

while(st!=0)

{

if(st%2==0)

{

n=n\*n;

st=st/2;

}

else

{

f=f\*n;

st=st-1;

}

counter++;

}

cout << i << " "<<counter<<endl;

counter=0;

cout << f << endl;

f=1;

n=2;

}

return 0;

}

**Конец листинга**