Абрашкина А.

**Лабораторная работа №1.**

**Оценка временной сложности алгоритмов**

Целью лабораторной работы является приобретение навыков исследования временной сложности алгоритмов и определения ее асимптотических оценок.

* + 1. Требования к содержанию, оформлению и порядку выполнения

В содержательной части отчета по выполнению лабораторной работы требуется привести описание алгоритма, выбранного согласно своему варианту, провести его анализ и определить асимптотические оценки его временной сложности. Алгоритм рекомендуется оформлять с помощью блок-схем.

* + 1. Теоретическая часть

Теоретические сведения, необходимые для выполнения лабораторной работы, представлены в лекции.

* + 1. Общая постановка задачи

Требуется провести анализ и оценку временной сложности заданного алгоритма. Варианты заданий представлены в таблице в следующем разделе.

В качестве дополнительных заданий рекомендуется программно реализовать заданный алгоритм.

* + 1. Список индивидуальных данных

Данные для выполнения лабораторной работы сведены в табл.Л2.1.

Таблица Л2.1.

Варианты заданий к лабораторной работе № 2

|  |  |
| --- | --- |
| *Вариант* | *Алгоритм* |
| 1 | Тривиальный алгоритм возведения в степень (рис. 3.1) |
| 2 | Рекурсивный алгоритм возведения в степень (рис. 3.2) |
| 3 | Алгоритм быстрого возведения в степень (рис. 3.3 а) |
| 4 | Алгоритм быстрого возведения в степень (рис. 3.3 б) |
| 5 | Алгоритм вычисления значения многочлена (рис. 3.4) |
| 6 | Алгоритм вычисления значения многочлена по схеме Горнера (рис. 3.5) |
| 7 | Алгоритм сортировки обменом (рис. 4.5) |
| 8 | Алгоритм сортировки выбором (рис. 4.7) |
| 9 | Алгоритм сортировки вставками (рис. 4.9) |
| 10 | Алгоритм быстрой сортировки (рис. 4.14, 4.16) |

* + 1. Вариант №2

**Ход работы:**

Требуется провести анализ и оценку временной сложности рекурсивного алгоритма возведения в степень. Алгоритм предназначен для решения следующей задачи: дано определение: f(n)=a^n: f(0) = 1; f(n) = f(n/2)^2, если n чётное число.

Напишите рекурсивную функцию, которая вычисляет f(n) согласно заданному определению.

 2 алгоритма вычисления f(n)=a^n: итеративный и рекурсивный. Оценка времени их работы

#include <stdio.h>

double power(int n) {

if ( n == 0 ) return 1;

return a \* power(a, n - 1);

}

#include <stdio.h>

double power(double a, int n) {

int i;

double x = 1;

for ( i = 1; i < n; i++) {

x \*= a;

}

return x;

}

Оба алгоритма работают линейно по n время.

#include <stdio.h>

double power(int n) {

if ( n == 0 ) return 1;

if ( n % 2 == 1)

return power(a\*a, n/2);

else

return a \* power(a, n - 1);

}

- эта рекурсия работает время O(log n). Оценка выводится, если нарисовать цепочку рекурсивных вызовов, представляя n в двоичной системе счисления.

Рекурсивный алгоритм возведения в степень основан на выше приведенных соображениях. Его блок-схема приведена на рис. Л2.1.

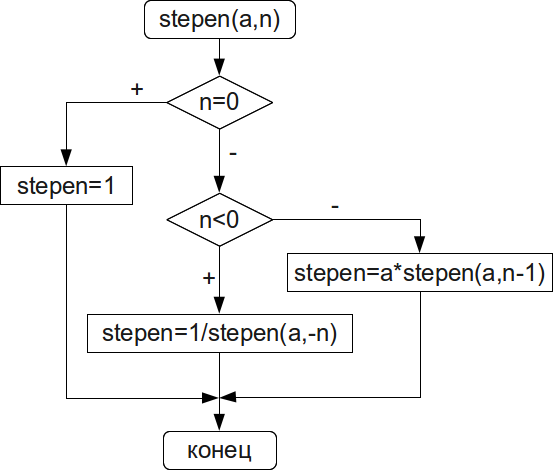


Рис. Л2.1. Рекурсивный алгоритм возведения в степень

Вывод. В лабораторной работе проведен анализ рекурсивного алгоритма возведения в степень. Его временная сложность имеет верхнюю оценку *О*(log*n*). Данный алгоритм имеет равную оценку временной сложности, как и итеративный алгоритм.

* + 1. Контрольные вопросы к защите

1. Понятие временной сложности алгоритма.
2. Определение асимптотических оценок временной сложности.
3. Основные принципы получения асимптотических оценок.
4. *Правила анализа алгоритмов с целью определения их временной сложности.*