

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем

Лабораторная работа №4

по дисциплине: Математическая логика и теория алгоритмов
тема: «Теория алгоритмов»

Вариант - 23

Выполнил: ст. группы ПВ-211
Тутов Данил Андреевич

Проверили:
Куценко Дмитрий Александрович
Рязанов Юрий Дмитриевич

Белгород 2022 г.

Цель работы: изучить теорию алгоритмов

Задания

Теоретическая часть:

Решить задачи из лабораторного практикума по математической логике согласно своему варианту. Для построения машин Тьюринга и Поста воспользоваться программным эмулятором Algo2000 или аналогичным.

Машина Тьюринга:

> - сдвиг вправо

> - сдвиг влево

- - остаться на месте

_ - пробел

№4

Дан номер года. Определить, високосный он или нет.

В начальный момент времени каретка расположена над первой цифрой номера, изначальное состояние Q1, алфавит: 1, 2, 3, 4, 6, 7, 8, 9, 0.

Если год високосный на ленте остаётся цифра 1, в ином случае - 0.

A/Q	Q0	Q1	Q2	Q4	Q5	Q6	Q7
1		>	_ < Q7	_ < Q6	_ < Q7	_ < Q6	_ < Q7
2		>	_ < Q4	_ < Q7	_ < Q6	_ < Q6	_ < Q7
3		>	_ < Q7	_ < Q6	_ < Q7	_ < Q6	_ < Q7
4		>	_ < Q5	_ < Q7	_ < Q6	_ < Q6	_ < Q7
5		>	_ < Q7	_ < Q6	_ < Q7	_ < Q6	_ < Q7
6		>	_ < Q4	_ < Q7	_ < Q6	_ < Q6	_ < Q7
7		>	_ < Q7	_ < Q6	_ < Q7	_ < Q6	_ < Q7
8		>	_ < Q5	_ < Q7	_ < Q6	_ < Q6	_ < Q7
9		>	_ < Q7	_ < Q6	_ < Q7	_ < Q6	_ < Q7
0		>	_ < Q5	_ < Q7	_ < Q6	_ < Q6	_ < Q7
Пробел_		< Q2				1 - Q0	0 - Q0
Комментарий:		Поиск конца	Обработка последней цифры	Обработка предпоследней цифры, если последняя 2 или 6	Обработка предпоследней цифры, если последняя 0, 4 или 8	Високосный год	Невисокосный год

Эмулятор машины Тьюринга

[illegible]

Слово: 24	Поместить	Очистить ленту
-----------	-----------	----------------

Алфавит: 1234567890	Задать
---------------------	--------

Начальное состояние: q1 v

Машина:

состояние (+)	1	2	3	4	5	6	7	8	9	0	λ	комментарий
q0 (-)	N	N	N	N	N	N	N	N	N	N	N	состояние
q1 (-)	R	R	R	R	R	R	R	R	R	R	, L, q2	ищем к
q2 (-)	λ, L, q7	λ, L, q4	λ, L, q7	λ, L, q5	λ, L, q7	λ, L, q4	λ, L, q7	λ, L, q5	λ, L, q7	λ, L, q5	N	Обработка
q4 (-)	λ, L, q6	λ, L, q7	λ, L, q6	λ, L, q7	λ, L, q6	λ, L, q7	λ, L, q6	λ, L, q7	λ, L, q6	λ, L, q7	N	Обработка
q5 (-)	λ, L, q7	λ, L, q6	λ, L, q7	λ, L, q6	λ, L, q7	λ, L, q6	λ, L, q7	λ, L, q6	λ, L, q7	λ, L, q6	N	Обработка
q6 (-)	λ, L, q6	λ, L, q6	λ, L, q6	λ, L, q6	λ, L, q6	λ, L, q6	λ, L, q6	λ, L, q6	λ, L, q6	λ, L, q6	1, N, q0	Год - не
q7 (-)	λ, L, q7	λ, L, q7	λ, L, q7	λ, L, q7	λ, L, q7	λ, L, q7	λ, L, q7	λ, L, q7	λ, L, q7	λ, L, q7	0, N, q0	Гол - не

№20

На информационной ленте машины Тьюринга находится массив из $2 \cdot N$ меток. Уменьшить этот массив в 2 раза.

В начальный момент времени каретка расположена над первой цифрой номера, изначальное состояние Q1, алфавит: 1, #.

1 - метка, # - символ, используемый при решении задачи.

A/Q	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
1		1 < Q2		1 > Q4	_ > Q3	_ > Q6		1 > Q7	1 < Q8	1 > Q9
#			# > Q3	# < Q5	# < Q5	_ > Q9	# >Q7		# < Q5	_ > Q0
—						_ < Q5	_ > Q6	1 < Q8		_ > Q9
Комментарий:		Ставим перед массивом меток #	Ставим перед массивом меток #	Удаляем каждую вторую метку в массиве, и ставим после массива меток #	Удаляем каждую вторую метку в массиве, и ставим после массива меток #	Перенос меток за последний символ #, удаление символа #, если все метки перенесены	Перенос меток за последний символ #	Перенос меток за последний символ #	Перенос меток за последний символ #	Удаление последнего символа #

<

λ

λ

λ

λ

λ

λ

λ

λ

λ

λ

λ

λ

λ

λ

λ

λ

λ

λ

λ

λ

1

1

1

λ

>

Слово:

Поместить

Очистить ленту

Алфавит:

Задать

Начальное состояние:

Машина:

Запустить

Сделать шаг

Сброс

Сохранить

Загрузить

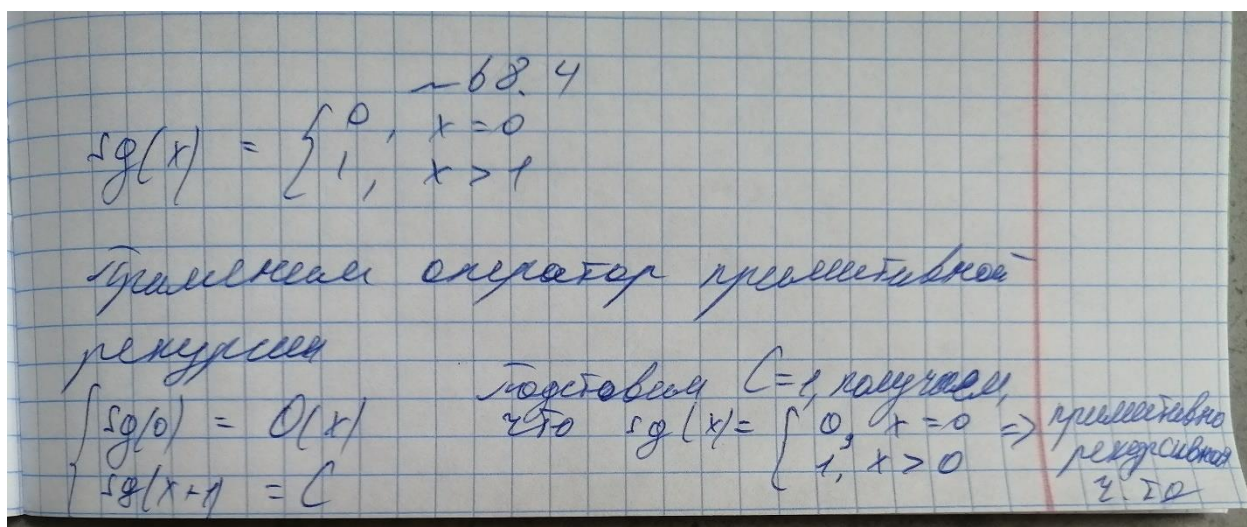
Состояние (+)		1	#	λ	комментарий
q0	⊖	N	N	N	
q1	⊖	1, L, q2	N	N	
q2	⊖	N	N	#, R, q3	
q3	⊖	1, R, q4	N	#, L, q5	
q4	⊖	, R, q3	N	#, L, q5	
q5	⊖	, R, q6	, R, q9	, L, q5	
q6	⊖	N	#, R, q7	, R, q6	
q7	⊖	1, R, q7	N	1, L, q8	
q8	⊖	1, L, q8	#, L, q5	N	
q9	⊖	1, R, q9	, R, q0	, R, q9	

№37

Число k представляется на ленте машины Поста $k + 1$ идущими подряд метками. Одна метка соответствует нулю. Составить программу копирования исходного числа и прибавления к нему единицы. Головка расположена над одной из меток, принадлежащих заданному числу k . При этом исходное число должно остаться на ленте.

Команд	Переход	Комментарий
<		Ищем свободную ячейку перед крайней левой меткой
?	3, 1	Ищем свободную ячейку перед крайней левой меткой
>		Переходим на крайнюю левую метку
?	17, 5	Переходим на крайнюю левую метку
0		Убираем метку(временно)
>		Ищем свободную ячейку после крайней правой метки
?	8, 6	Ищем свободную ячейку после крайней правой метки
>		Пропускаем 1 свободную ячейку, чтобы разделить изначальный и конечный массивы меток
?	10, 8	Ищем куда поставить метку в результирующий массив: через пробел после начального массива, либо после уже поставленной метки результирующего массива
1		Ставим метку в результирующий массив
<		Ищем пробел, разделяющий массивы
?	13, 11	Ищем пробел, разделяющий массивы
<		Ищем пробел в начальном массиве, откуда брали метку, либо же доходим до начала изначального массива, если перенесли только одну метку
?	15, 13	Ищем пробел в начальном массиве, откуда брали метку, либо же доходим до начала изначального массива, если перенесли только одну метку
1		Вставляем убранную ранее метку

№68.4 Доказать, что следующие функции примитивно рекурсивны:



Практическая часть

Практическая часть: разработать программу на языке Brainfuck, решающую задачу «На информационной ленте машины Тьюринга в трех ячейках в произвольном порядке записаны три различные буквы “А” (здесь и далее без кавычек), “В” и “С”. Необходимо составить функциональную схему машины Тьюринга, которая сумеет поменять местами крайние буквы».

- Считываем. Ввод заканчиваем 0.
 $>,[>,<[-]$
- Идем в начало.
 $<[<]>$
- Копируем первую букву в соседнюю ячейку, исходную букву удаляем.
 $[-<+>]$
- Идем в конец и копируем последнюю букву в первую.
 $>>[-<<+>>]$
- Идем в самое начало и копируем содержимое в последнюю ячейку.
 $<<<[->>>+<<<]$
- Вывод результата.
 $>[.>]$

Программная реализация

```
#include <iostream>
#include <windows.h>
#include <ctime>

void read_bfFile(FILE *cmdlist) {
    char c;
    int i = 0;
    while (!feof(cmdlist)) {
        if (i == 50) {
            printf("\n");
            i = 0;
        }
        fscanf(cmdlist, "%c", &c);
        if (!feof(cmdlist)) {
            printf("%c", c);
            i++;
        }
    }
    printf("\n");
}

void translate_brainfuck_fFile(FILE *cmdlist, char *tier) {
    int loop;
    char cmd_char;

    do {
        fscanf(cmdlist, "%c", &cmd_char);
        switch (cmd_char) {
            case '>':
                tier++;
                break;
            case '<':
                tier--;
                break;
            case '+':
                (*tier)++;
                break;
            case '-':
                (*tier)--;
                break;
            case '.':
                putchar(*tier);
                break;
            case ',':
                *tier = getchar();
                break;
            case '[':
                if (!*tier) {
                    loop = 1;
                    while (loop) {
                        fscanf(cmdlist, "%c", &cmd_char);
                        if (cmd_char == '[') {
                            loop++;
                        }
                        if (cmd_char == ']') {
                            loop--;
                        }
                    }
                }
                break;
        }
    } while (1);
}
```

```

        case ']':
            if (*tier) {
                loop = 1;
                while (loop) {
                    fseek(cmdlist, -2, SEEK_CUR);
                    fscanf(cmdlist, "%c", &cmd_char);
                    if (cmd_char == '[') {
                        loop--;
                    }
                    if (cmd_char == ']') {
                        loop++;
                    }
                }
                break;
            }
            default:
                fprintf(stderr, "Error");
                exit(1);
        }
        (*tier)++;
    } while ((!feof(cmdlist)));

    printf("\n");
}

int main() {
    SetConsoleOutputCP(CP_UTF8);
    srand(time(NULL));

    char cpu[30000];
    FILE *read = fopen("exercise.txt", "r");
    read_bfFile(read);
    translate_brainfuck_fFile(read, cpu);
}

```

Вывод: в ходе лабораторной работы была изучена теория алгоритмов