Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе № 4

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Бинарное дерево поиска»

Выполнили

студенты группы 22ВВВ2:

Зубриянова А.А.

Кондратьева В.И.

Приняли:

Акифьев И.В.

Митрохин М.А.

Пенза 2023

**Название**

Бинарное дерево поиска.

**Цель работы**

Изучить работу бинарного дерева поиска.

**Лабораторное задание**

Задание 1: реализовать алгоритм поиска вводимого с клавиатуры значения в уже созданном дереве.

Задание 2: реализовать функцию подсчёта числа вхождений заданного элемента в дерево.

Задание 3: \* изменить функцию добавления элементов для исключения добавления одинаковых символов.

Задание 4: \* оценить сложность процедуры поиска по значению в бинарном дереве.

**Описание метода решения задачи**

Задание 1 и 2:

Для поиска значения в уже созданном дереве используется функция FindTree. Она проходит все дерево и при нахождении искомого элемента инкрементирует переменную С (С – это подсчет количества вхождений числа).

Задание 3:

Вызывается функция FindTree, изменяющая значение переменной С. С = 1, если текущий элемент дерева совпадает с элементом, который уже встречался ранее.

Функция создания элемента дерева CreateTree вызывается при условии, что С == 0, что исключает создание одинаковых элементов.

Задание 4:

Сложность процедуры поиска по значению : O(n). Подсчет количества рекурсий осуществляет переменная R, инкрементирующаяся в функции FindTree.

**Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

//struct Node\* root;

int C;

int R;

void FindTree(struct Node\* root, int data);

struct Node\* CreateTree(struct Node\* root, struct Node\* r, int data);

void print\_tree(struct Node\* r, int l);

struct Node

{

int data;

struct Node\* left;

struct Node\* right;

};

int main()

{

setlocale(LC\_ALL, "");

int D;

struct Node\* root;

root = NULL;

printf("-1 - окончание построения дерева\n");

while (1)

{

printf("Введите число: ");

scanf("%d", &D);

if (D == -1)

{

printf("Построение дерева окончено\n\n");

break;

}

else

{

C = 0; R = 0;

FindTree(root, D); // функцию для исключения добавления одинаковых символов

if (C == 0)

{

root = CreateTree(root, root, D);

}

else

{

printf("Введите другое число\n");

}

}

print\_tree(root, 0);

}

printf("-1 - окончание поиска\n");

while (1)

{

printf("Введите искомое число: ");

scanf("%d", &D);

if (D == -1)

{

printf("Поиск окончен\n\n");

break;

}

else

{

C = 0; R = 0;

FindTree(root, D);

printf("Число вхождений = %d\n", C);

printf("Сложность поиска = %d\n", R);

}

}

return 0;

}

void FindTree(struct Node\* root, int data)

{

R++;

if (root == NULL) return;

if (data == root->data)

{

C++;

}

if (data < root->data)

FindTree(root->left, data);

else

FindTree(root->right, data);

}

struct Node\* CreateTree(struct Node\* root, struct Node\* r, int data)

{

//printf("(%lld %lld %d)\n", root, r, data);

if (r == NULL)

{

r = (struct Node\*)malloc(sizeof(struct Node));

if (r == NULL)

{

printf("Ошибка выделения памяти");

exit(1);

}

r->left = NULL;

r->right = NULL;

r->data = data;

if (root == NULL) return r;

if (data < root->data) root->left = r;

else root->right = r;

return r;

}

if (data < r->data)

CreateTree(r, r->left, data);

else

CreateTree(r, r->right, data);

return root;

}

void print\_tree(struct Node\* r, int l)

{

if (r == NULL)

{

return;

}

print\_tree(r->right, l + 1);

printf("%d-", l);

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

{

printf(" ");

}

printf("%d\n", r->data);

print\_tree(r->left, l + 1);

}

**Результаты работы программы**

Результаты работы программы представлены на рисунках:

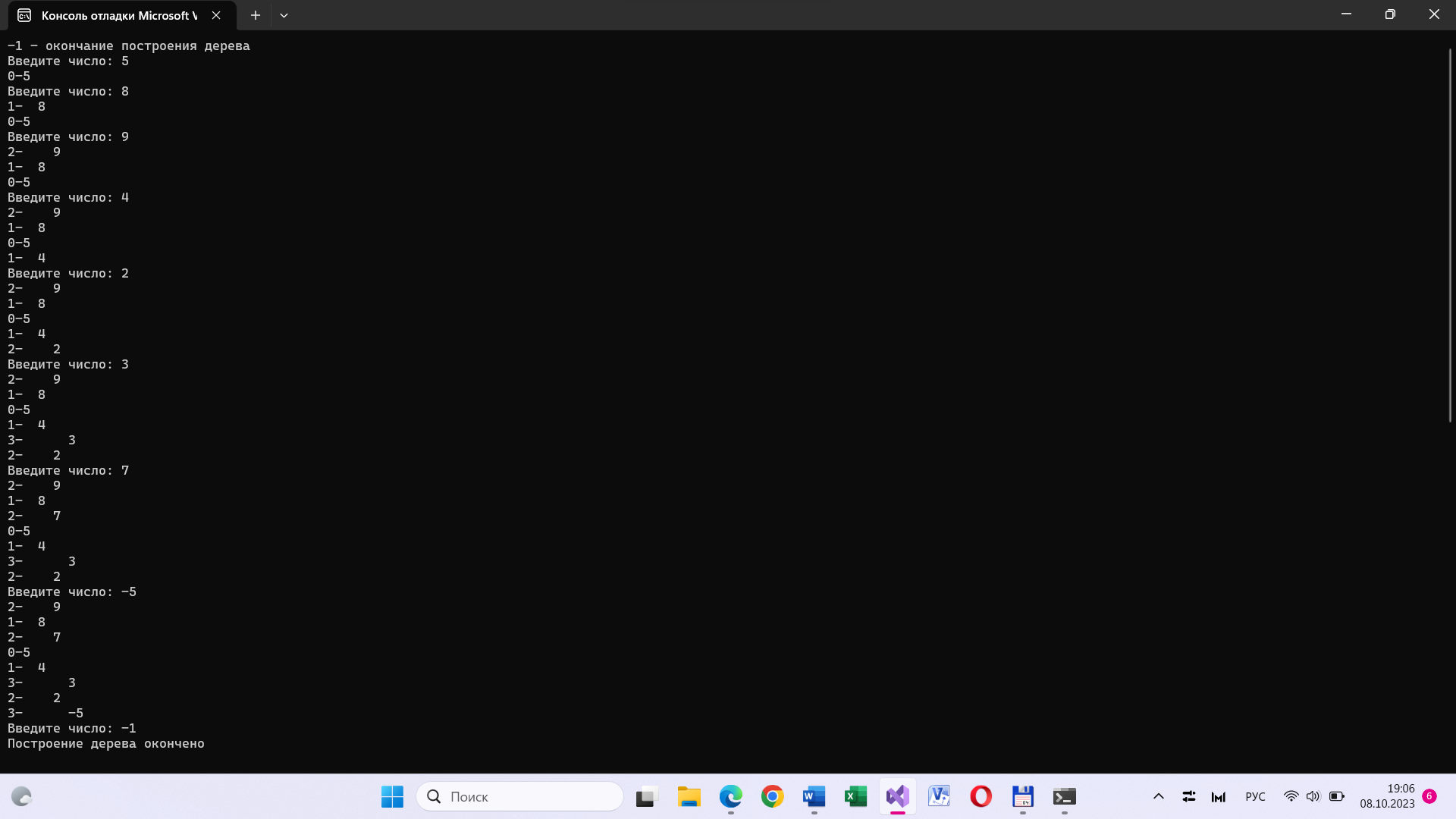


Рисунок 1 – Создание дерева

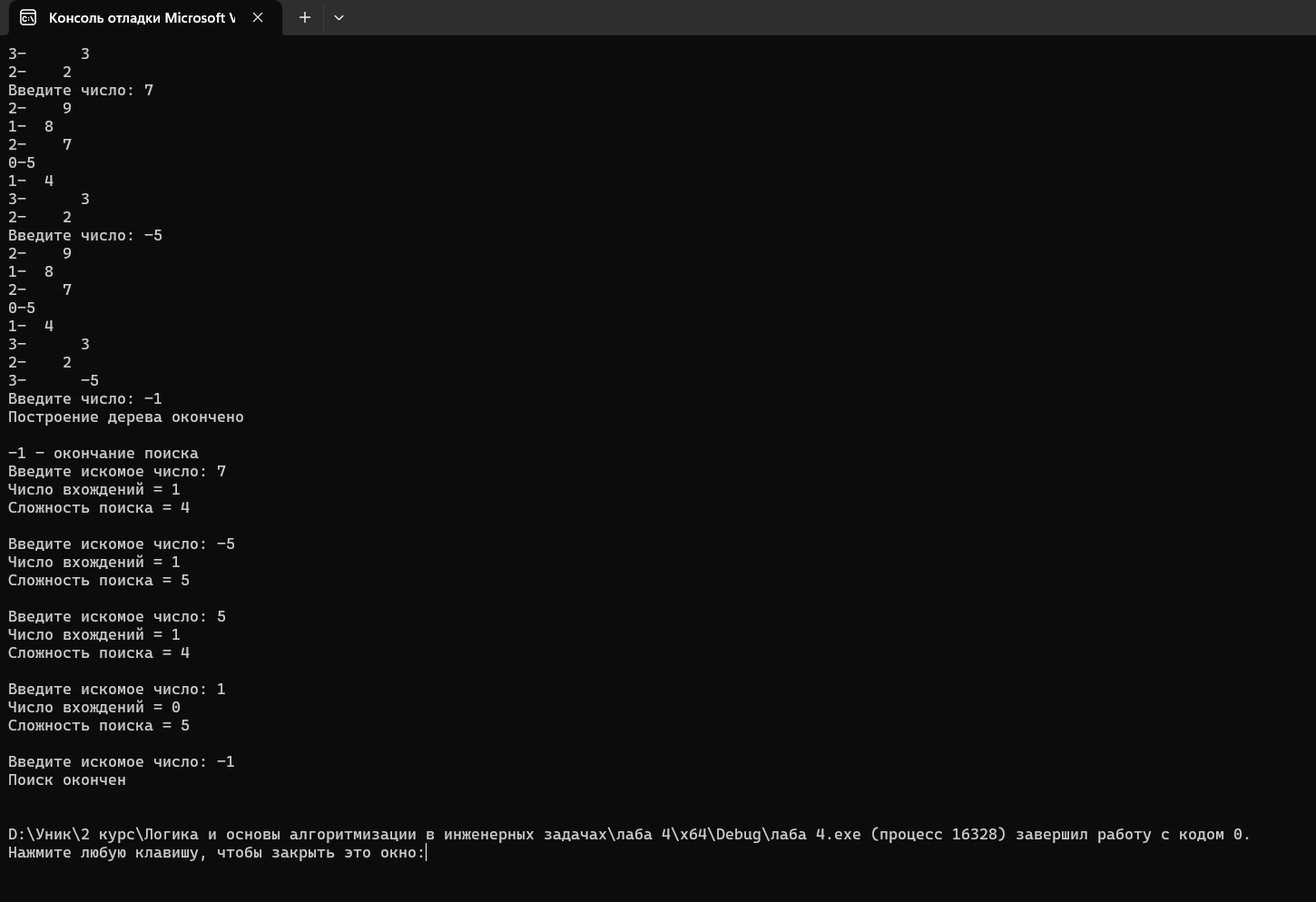


Рисунок 2 – Поиск элементов, число вхождений, сложность поиска

**Вывод**

В ходе выполнения лабораторной работы была разработана программа, выполняющая операции в бинарном дереве поиска. Результаты работы программы совпали с ожидаемыми результатами, следовательно, программа работает без ошибок.

Получили опыт в создании проектов в среде Microsoft Visual Studio, приобрели навыки программирования алгоритмов.