



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ

Информатика и системы управления

КАФЕДРА

Программное обеспечение ЭВМ и информационные технологии

## **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**

Студент

Кашима Ахмед

Группа

ИУ7 – 33Б

2022 г.

## **Вариант № 5**

### **ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ**

Построить дерево в соответствии со своим вариантом задания. Вывести его на экран в виде дерева. Реализовать основные операции работы с деревом: обход дерева, включение, исключение и поиск узлов. Сравнить эффективность алгоритмов сортировки и поиска в зависимости от высоты деревьев и степени их ветвления.

### **ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ**

Построить частотный словарь (слово – количество повторений) из слов текстового файла в виде дерева двоичного поиска. Вывести его на экран в виде дерева. Осуществить поиск указанного слова в дереве и в файле. Если слова нет, то (по желанию пользователя) добавить его в дерево и, соответственно, в файл. Сравнить время поиска слова в дереве и в файле.

#### **Входные данные:**

1. **Целое число, представляющее собой номер команды:**

целое число в диапазоне от 0 до 7.

#### **Выходные данные:**

Результат выполнения определенной команды.

### **ОПИСАНИЕ СТРУКТУРЫ ДАННЫХ**

*// Структура дерева*

```
typedef struct tree_node
{
    const char *name; // название узла
    struct tree_node *left; // меньшие
    struct tree_node *right; // старшие
}tree_node_t;
```

### **Меню программы:**

- 1 - Прочитать дерево из файла.
- 2 - Вывести дерево на экран.
- 3 — Добавить узел в дерево.
- 4 — Найти слово в дереве.
- 5 — Удалить слово из дерева.
- 6 - Сравнить время поиска слова в дереве и в файле..
- 7 - Сравнить время сортировки слов в дереве и в файле.
- 0 – Выйти из программы.

### **Обращение к программе:**

Запускается через терминал с помощью команды ./app.exe.

### **Аварийные ситуации:**

1. Некорректный ввод номера команды.

На входе: число, большее чем 7 или меньшее, чем 0.

На выходе: сообщение «Ошибка: Вы выбрали неправильное действие!»

2. Совершение вывод дерева, сравнения поиска слова и поиск в дереве при пустом дереве.

На входе: целое число в диапазоне от 2 до 7 (номер команды, кроме команды 3).

На выходе: сообщение «Ошибка: Дерево пустое!»

## ОПИСАНИЕ АЛГОРИТМА

Алгоритм поиска элемента в дереве : Обходятся все вершины дерева начиная с корня до левого поддерева потом до правого поддерева пока не встретим вершину, которую хотим найти.

Алгоритм обхода дерева в префиксной форме : Обходятся корень до поддереваев.(Сначала обходим левое поддерево, потом правое)

Алгоритм удаления элемента в дереве : Обходятся все вершины дерева в префиксную форму, и если нашли вершину которую хотим удалить, то мы ее удаляем и на это место записываем самый маленький элемент правого дерева слева.

## НАБОР ТЕСТОВ

	Название теста	Пользовательский ввод	Результат
1	Некорректный ввод команды	-1	Ошибка: Вы выбрали неправильное действие!
2	Удаление элемента из пустого дерева	Команда 5	Ошибка: Дерево пустое!
4	Поиск вершины в	Команда 4	Ошибка: Дерево

	пустое дерево		пустое!
5	Время поиска слова в пустое дерево	Команда 6	Ошибка: Дерево пустое!
6	Время сортировки вершин пустого дерева	Команда 7	Ошибка: Дерево пустое!

## ОЦЕНКА ЭФФЕКТИВНОСТИ АЛГОРИТМА ПОИСКА

Измерения поиска будет производиться в секундах.

Размер	Дерево	Файл
23	0.000003	0.000043
100	0.000004	0.000054
200	0.000006	0.000064
500	0.000005	0.000086

## ОЦЕНКА ЭФФЕКТИВНОСТИ АЛГОРИТМА СОРТИРОВКИ

Измерения сортировки будет производиться в секундах.

Размер	Дерево	Файл
23	0.000023	0.000086
100	0.000053	0.000171
200	0.000094	0.000281
500	0.000215	0.000518

Из данных таблиц можно сделать вывод, что алгоритм поиска и алгоритм сортировки дерева лучше работает, чем алгоритм поиска и сортировки файла.

Алгоритм сортировки дерева лучше работает потому, что в дереве все элементы уже отсортированы и, те элементы которые повторяются не записываются в дерево.

## ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

### 1. Что такое дерево?

Дерево – нелинейная структура данных, которая используется для представления иерархических связей «один ко многим». Дерево с базовым типом T определяется рекурсивно: это либо пустая структура (пустое дерево), либо узел типа T с конечным числом древовидных структур того же типа – поддеревьев.

### 2. Как выделяется память под представление деревьев?

Выделение памяти под деревья определяется типом их представления. Это может быть таблица связей с предками (No вершины - No родителя), или связный список сыновей. Оба представления можно реализовать как с помощью матрицы, так и с помощью списков. При динамическом представлении деревьев (когда элементы можно удалять и добавлять) целесообразнее использовать списки - т.е. выделять память под каждый элемент динамически.

3. Какие бывают типы деревьев?

АВЛ-деревья, сбалансированные деревья, двоичные, двоичного поиска.

4. Какие стандартные операции возможны над деревьями?

Основные операции с деревьями: обход (инфиксный, префиксный, постфиксный), поиск элемента по дереву, включение и исключение элемента из дерева.

5. Что такое дерево двоичного поиска?

Дерево двоичного поиска - дерево, в котором все левые потомки «моложе» предка, а все правые - «старше». Это свойство выполняется для любого узла, включая корень.

## **Вывод**

Основным преимуществом двоичного дерева перед другими структурами данных является возможная высокая эффективность реализации основанных на нём алгоритмов поиска и сортировки. Хранение и обработка дерева требует аккуратного обращения с памятью.