	<p>Министерство образования и науки 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)</p>
---	---

ФАКУЛЬТЕТ _____ Информатика и системы управления
 КАФЕДРА _____ Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 6
По дисциплине «Типы и структуре данных»

Название «Деревья»

Студент Дубов Андрей Игоревич
 фамилия, имя, отчество

Группа ИУ7-33Б

Вариант 5

Тип лабораторной работы Учебная

Студент	_____	<u>Дубов А. И.</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>
Преподаватель	_____	<u>Рыбкин Ю. А.</u>
	<i>подпись, дата</i>	<u>Силантьева А. В.</u>
		<i>фамилия, и.о.</i>

2022 г.

Оглавление

Условие задачи.....	3
Описание технического задания.....	3
Входные данные:.....	3
Выходные данные:.....	3
Аварийные ситуации:.....	3
Описание структуры данных.....	3
Описание алгоритма.....	4
Набор тестов.....	4
// тесты.....	4
Оценка эффективности.....	4
Вывод.....	4
Ответы на контрольные вопросы.....	4

Условие задачи

Построить дерево в соответствии с заданным 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 дерева и степени его 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:

Файл с текстом.

Выходные данные:

Дерево, информация о частотности слова.

Аварийные ситуации:

1. Некорректный ввод номера команды.

Описание структуры данных

Структура стека

```
typedef struct tree
{
    char word[100];
    int key;
    struct tree *left;
    struct tree *right;
    struct tree *parent;
} node_t;
```

word – слово ветки

key – количество повторений

left – указатель на левое поддерево

right – указатель на правое поддерево

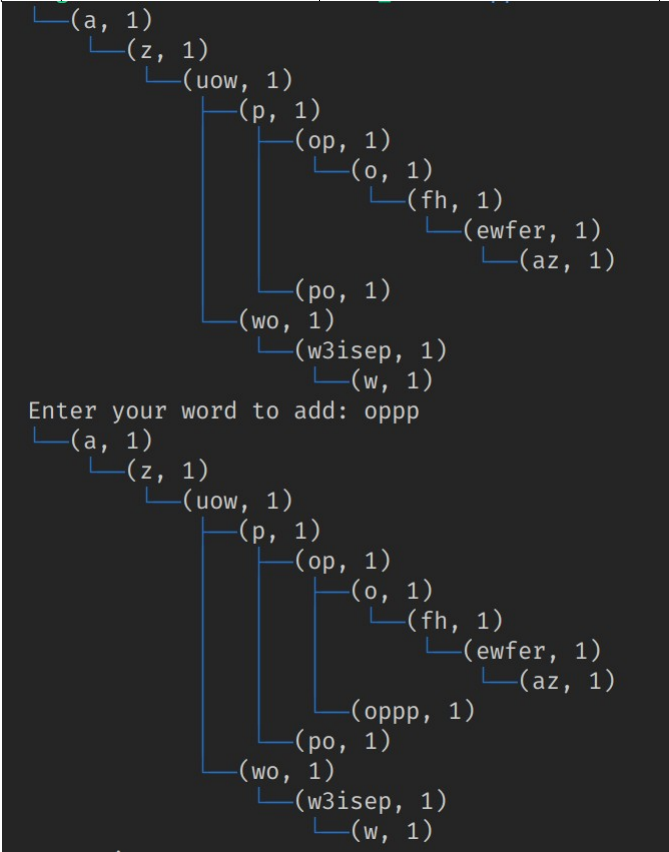
parent – указатель на поддерево родителя

Описание алгоритма

- 1. Пользователю предлагается режим
- 2. Пользователь управляет выделенной памятью
- 3. Пока пользователь не введет 0 (выход из программы), ему будет предложено вводить номера команд и выполнять действия по выбору.

Набор тестов

	Название теста	Пользователь вводит	Вывод
1	Некорректный ввод команды	45	No such option or wrong input
2	Пустой ввод	Пустой ввод.	No such option or wrong input
3	Команда 0	0	Выход из программы



Оценка эффективности

В среднем поиск по дереву занимает 1433 тика, а по файлу 20349 тика. Проводилось 50 измерений.

Вывод

Деревья удобны, когда требуется долго и по несколько раз искать данные. Они нагляднее в представлении, но расходуют память, так как хранят много указателей. Поиск по файлу в разы дольше, чем по дереву, поскольку при поиске по файлу делается проход по всему файлу.

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 (№ вершины - № 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?

N-арное дерево, сбалансированное дерево, бинарное дерево, бинарное дерево поиска, хвойное, красно-чёрное дерево, лиственное

4. Какие стандартные операции возможны над деревьями?

Обход, поиск, добавление и удаление элемента.

5. Что такое дерево двоичного поиска?

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