

# Задача №6

## Деревья

**Цель работы – получить навыки применения двоичных деревьев**

**Задание: 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.**

### № варианта – № % 7

0. Построить двоичное дерево поиска из букв вводимой строки. Вывести его на экран в виде дерева. Выделить цветом все буквы, встречающиеся более одного раза. Удалить из дерева эти буквы. Вывести оставшиеся элементы дерева при постфиксном его обходе. Сравнить время удаления повторяющихся букв из дерева и из строки.

1. В текстовом файле содержатся целые числа. Построить двоичное дерево из чисел файла. Вывести его на экран в виде дерева. Определить количество узлов дерева на каждом уровне. Добавить число в дерево и в файл. Сравнить время добавления чисел в указанные структуры.

2. Построить двоичное дерево поиска, в вершинах которого находятся слова из текстового файла. Вывести его на экран в виде дерева. Определить количество вершин дерева, содержащих слова, начинающиеся на указанную букву. Выделить эти вершины цветом. Сравнить время поиска начинающихся на указанную букву слов в дереве и в файле.

3. Построить бинарное дерево поиска, в вершинах которого находятся слова из текстового файла (предполагается, что исходный файл не содержит повторяющихся слов). Вывести его на экран в виде дерева. Удалить все слова, начинающиеся на указанную букву. Сравнить время удаления слов, начинающихся на указанную букву, в дереве и в файле.

4. В файловой системе каталог файлов организован в виде бинарного дерева. Каждый узел обозначает файл, содержащий имя и атрибуты файла, в том числе и дату последнего обращения к файлу. Написать программу, которая обходит дерево и удаляет из него все файлы, последнее обращение к которым происходило до определенной даты. Вывести исходные и измененные деревья в виде дерева. Сравнить время удаления в дереве, построенном по алфавиту, со временем перестроения дерева по дате обращения и удаления в нем.

5. 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.

6. Каждый узел дерева содержит информацию о студенте: Фамилия, средний балл. Сформировать дерево бинарного поиска по фамилии. Написать программу обхода дерева и удаления всех студентов со средним баллом ниже или равно 2.0. Вывести исходное дерево и дерево после удаления. Сравнить время удаления в дереве, построенном по фамилии со временем перестроения по баллу и удаления в нем.