

## 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 – № по списку Mod 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. Используя подпрограмму, определить количество узлов дерева на каждом уровне. Добавить число в дерево и в файл. Сравнить время добавление 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 идеально сбалансированное бинарное дерево, в вершинах которого находятся слова их текстового файла. 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 исходное и измененное деревья в виде дерева.
5. Построить частотный словарь 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. Ввести значения переменных: от A до I. Построить и вывести на экран бинарное дерево следующего выражения:  $A + (B * (C + (D * (E + F) - (G - H)) + I))$ . Написать процедуры постфиксного, инфиксного и префиксного обхода дерева и вывести соответствующие выражения на экран. Подсчитать результат. Используя «польскую» запись, ввести данное выражение в стек. Сравнить время вычисления выражения с использованием дерева и стека.