

# Лабораторная работа № 4б «Работа с самобалансирующимися деревьями поиска»

## Введение

В процессе выполнения лабораторной работы необходимо решить следующие задачи:

1. Спроектировать и разработать на языке C:
  - (a) Прикладную программу, осуществляющую интерактивную работу пользователя с деревом.
  - (b) Библиотеку, предоставляющую функциональность по работе с деревом поиска, в соответствии с условиями индивидуального задания.
2. Выполнить таймирование (или профилирование) программы. Построить графики зависимости времени выполнения операций, предусмотренных индивидуальным заданием, от количества элементов в дереве.
3. Оценить сложность реализованных алгоритмов.

## Дополнительное задание

Необходимо выбрать один из представленных ниже вариантов, исходя из особенностей индивидуального задания:

1. *Особенности индивидуального задания:* ключ — строка, значение — число, дублирование ключей — разрешено.  
*Формулировка:* быстрый поиск слова в текстовом файле.  
*Возвращаемое значение:* вектор, состоящий из номеров строк, в которых присутствует искомое слово.
2. *Особенности индивидуального задания:* ключ — строка, значение — число, дублирование ключей — запрещено.  
*Формулировка:* быстрый поиск слова в текстовом файле.  
*Возвращаемое значение:* номер строки первого вхождения искомого слова.
3. *Особенности индивидуального задания:* ключ — строка, значение — строка, дублирование ключей — разрешено.  
*Формулировка:* быстрый поиск слова в текстовом файле.  
*Возвращаемое значение:* вектор, состоящий из строк с расположением искомого слова в формате <имя\_файла>:<номер\_строки>:<смещение\_в\_строке>.

4. *Особенности индивидуального задания:* ключ — строка, значение — строка, дублирование ключей — запрещено.

*Формулировка:* быстрый поиск слова в текстовом файле.

*Возвращаемое значение:* строка формата <имя\_файла>:<номер\_строки>:<смещение\_в\_строке>, указывающая на первое вхождение искомого слова.

5. *Особенности индивидуального задания:* ключ — число, значение — число, дублирование ключей — разрешено.

*Формулировка:* быстрый поиск числа в текстовом файле, строки которого содержат ноль или более чисел, разделённых запятой.

*Возвращаемое значение:* вектор, состоящий из номеров строк, в которых присутствует искомое число.

6. *Особенности индивидуального задания:* ключ — число, значение — число, дублирование ключей — запрещено.

*Формулировка:* быстрый поиск числа в текстовом файле, строки которого содержат ноль или более чисел, разделённых запятой.

*Возвращаемое значение:* номер строки первого вхождения искомого числа.

7. *Особенности индивидуального задания:* ключ — число, значение — строка, дублирование ключей — разрешено.

*Формулировка:* быстрый поиск числа в текстовом файле, строки которого содержат ноль или более чисел, разделённых запятой.

*Возвращаемое значение:* вектор, состоящий из строк с расположением искомого числа в формате <имя\_файла>:<номер\_строки>:<номер\_числа\_в\_строке>.

8. *Особенности индивидуального задания:* ключ — число, значение — строка, дублирование ключей — запрещено.

*Формулировка:* быстрый поиск числа в текстовом файле, строки которого содержат ноль или более чисел, разделённых запятой.

*Возвращаемое значение:* строка формата <имя\_файла>:<номер\_строки>:<номер\_числа\_в\_строке>, указывающая на первое вхождение искомого числа.

## **Операции, поддерживаемые деревом**

В программе необходимо предусмотреть возможность проведения следующих операций над деревом, особенности реализации которых определяются индивидуальным заданием:

1. добавление нового элемента;
2. удаление элемента;
3. обход;
4. поиск элемента по ключу;
5. специальный поиск элемента.

Кроме того, должны быть реализованы следующие общие операции:

1. форматированный вывод дерева «в виде дерева»;

2. графический вывод дерева при помощи локальной внешней утилиты или библиотеки (например, graphviz);
3. загрузка дерева из текстового файла следующего формата:
  - Key1
  - Info1
  - Key2
  - Info2
  - ...

## Примечания

1. Логически законченные части алгоритма решения задачи должны быть оформлены в виде отдельных функций с параметрами. Использование глобальных переменных не допускается.
2. Функции для работы с деревом должны быть организованы в виде отдельной библиотеки, которая используется основной программой.
3. Функции для работы с деревом не должны быть диалоговыми, т. е. они должны принимать все необходимые данные в качестве параметров и возвращать результат работы в виде соответствующих структур данных и кодов ошибок (исключение: функции вывода дерева).
4. Диалоговые функции должны использовать описанные выше функции.
5. Программа должна осуществлять проверку корректности вводимых данных.
6. В случае возникновения ошибочных ситуаций при выполнении операций с деревом программа должна выводить соответствующие сообщения, после чего продолжать работу.
7. Для сборки программы и библиотек должна использоваться система сборки (например: Make или CMake).
8. Библиотеки и прикладные программы должны собираться независимо друг от друга.
9. Программа должна корректным образом работать с памятью, для проверки необходимо использовать соответствующие программные средства, например: valgrind, санитайзеры, встроенные в IDE средства и т.д.