

Лабораторная работа №4а «Работа с двоичным деревом поиска»

Введение

В процессе выполнения лабораторной работы необходимо решить следующие основные задачи:

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

Дополнительное задание

При помощи реализованной библиотеки, осуществляющей работу с двоичным деревом поиска, написать программу для подсчета частоты встречаемости каждого слова в текстовом файле (если ключ — строка) или каждого числа в бинарном файле (если ключ — число).

Результат работы программа должна выводить в текстовый файл в формате «ключ — количество».

Результат не должен зависеть от верхнего/нижнего регистра в словах и от пунктуационных знаков.

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. При выборе способов реализации операций над деревом предпочтение следует отдавать не рекурсивным алгоритмам. Выбор рекурсивного алгоритма необходимо обосновать.
8. Для сборки программ и библиотек должна использоваться система сборки (например: Make или CMake).
9. Библиотеки и прикладные программы должны собираться независимо друг от друга.
10. Программа должна корректным образом работать с памятью, для проверки необходимо использовать соответствующие программные средства, например: valgrind, санитайзеры, встроенные в IDE средства и т.д.