

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

Введение

Задачи

В процессе выполнения лабораторной работы необходимо решить следующие задачи:

1. Спроектировать и разработать программу, осуществляющую интерактивную работу пользователя с двоичным деревом поиска.
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.

Дополнительные задачи

Существует ряд дополнительных задач, не обязательных к выполнению, но позволяющих получить дополнительные баллы.

За задачу, отмеченную «*», можно получить до 2 баллов по десятибалльной шкале, а за задание, отмеченное «**» — до 3 баллов по десятибалльной шкале.

1. * Реализовать графический вывод дерева при помощи локальной внешней утилиты или библиотеки (например, graphviz). При этом, отображение дерева в графическом интерфейсе пользователя или генерация файла с изображением должно происходить автоматически, без выполнения действий вручную со стороны пользователя.
2. ** 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.

Основные операции

В программе необходимо предусмотреть возможность проведения следующих операций на дереве, 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. 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, т. е. должен быть реализован принцип Model-View-Controller (MVC).
5. Программа должна осуществлять проверку корректности вводимых данных и, в случае ошибок, выдавать соответствующие сообщения, после чего продолжать работу.
6. В случае возникновения ошибочных ситуаций при выполнении операций с деревом программа должна выводить соответствующие сообщения, после чего продолжать работу.
7. Программа должна корректным образом работать с памятью, для проверки необходимо использовать соответствующие программные средства, например: valgrind, санитайзеры, встроенные в IDE средства и т.д.

Вариант №72

Основные параметры дерева

Поддерживаемые типы данных

Должны поддерживаться следующие типы данных:

- ключ — ноль-терминированная строка произвольной длины;
- значение — ноль-терминированная строка произвольной длины.

Содержимое узла

Узел дерева должен содержать:

- ключ;
- указатели на правое и левое поддеревья;
- указатель на информационное поле.

Принцип работы с дублирующимися ключами

В дереве могут храниться записи с дублирующимися ключами. Элементы с дублирующимися ключами должны храниться непосредственно в дереве, их добавление должно осуществляться путём удлинения его ветвей.

Операции, поддерживаемые деревом

Добавление нового элемента

Добавление нового элемента в дерево без нарушения свойств упорядоченности. Если запись с данным ключом уже присутствует в дереве, а дублирование ключей не допускается, то необходимо обновить значение информационного поля существующей записи, а старое вернуть в качестве результата.

Удаление элемента

Удаление элемента, заданного своим ключом, без нарушения свойств упорядоченности дерева. Если в дереве присутствуют несколько элементов с указанным ключом, то необходимо удалить наиболее старый из них.

Обход дерева

Вывод всего содержимого дерева в прямом порядке следования ключей, не входящих в заданный диапазон.

Поиск элемента по ключу

Поиск информации по заданному ключу. Если элементов с требуемым значением ключа может быть несколько, то необходимо в качестве результата вернуть их все. Возврат необходимо осуществлять при помощи вектора указателей, возврат копий элементов не допускается.

Специальный поиск элемента

Поиск элемента с наибольшим значением ключа, не превышающим заданное (если таких элементов несколько — действовать по аналогии с операцией поиска по ключу).