

Лабораторная работа № 2 по курсу дискретного анализа: Словарь

Выполнил студент группы М08-207 МАИ Ваньков Денис.

Условие

1. Необходимо создать программную библиотеку, реализующую указанную структуру данных, на основе которой разработать программу - словарь. В словаре каждому ключу, представляющему из себя регистронезависимую последовательность букв английского алфавита длиной не более 256 символов, поставлен в соответствие некоторый номер, от 0 до $2^{64} - 1$. Разным словам может быть поставлен в соответствие один и тот же номер.

Программа должна обрабатывать строки входного формата до его окончания.

2. Вариант задания 4. Тип дерева: В - дерево.

Метод решения

Программа зачитывает пары (ключ, значение), затем проверяет есть ли такой элемент в дереве, если такого элемента нет, то с помощью функции вставки в дерево добавляет его, если он существует, то программа выводит 'Exist'. В процессе добавления элементов постоянно происходит проверка на удовлетворение всем условиям В - дерева т. е:

1. Каждый узел содержит хотя бы один ключ. Корень содержит от 1 до $2t-1$ ключей. Любой другой узел содержит от $t-1$ до $2t-1$ ключей. Листья не являются исключением из этого правила. Здесь t — параметр дерева, не меньший 2.
2. У листьев потомков нет. Любой другой узел, содержащий n ключей, содержит $n+1$ потомков. При этом: а) Ключ предка больше ключей потомков. б) Для каждого соседа в вершине правый ключ больше левого, а также ключи правой вершины больше любого из левой.
3. Глубина всех листьев одинакова.

Описание программы

Код разбит на несколько файлов. Файл библиотеки *TString.h*, в котором описан класс строк, файл *TString.cpp*, в котором описаны функции класса *TString*. В заголовочном файле *TBTree.h* описан класс В - дерева, а в *TBTree.cpp* все функции этого класса, такие как: добавление элемента по ключу, поиск ключа в дереве, удаление из дерева, балансировки дерева, сдвиги, и починка узла (функция соблюдающая условие построения дерева), а также функции сохранения дерева и его загрузки. Файл *main.cpp*, в котором производится считывание данных, определение операции, и выполнение определенной операции в зависимости от входных данных. Все файлы собираются при помощи *Makefile*.

Дневник отладки

В процессе отладки я вводил различные операции и ключи и проверял правильно ли они выполняются. Также проверял, на ключах большой длины и ключах с большими значениями.

При вставке элемента происходит проверка на существование его в дереве, и при его отсутствии последующая вставка, которая производится путем сравнения ключей, из - за чего элемент вставляется в нужное место. Параллельно происходят проверки на условие существования дерева, если хоть одно условие не выполняется дерево начинает перестраиваться при помощи вспомогательных функций.

Удаление происходит по алгоритму, аналогичному вставке: производится проверка на существование, и в случае существования элемент удаляется, параллельно производятся проверки на условие существования дерева.

Сохранение и загрузка дерева производится за линейное время. Происходит рекурсивно, постепенно заполняя сначала потомков, а затем и предков.

Выводы

Данный алгоритм может быть использован для хранения численных данных большого размера с помощью ключей - слов в "неком" словаре, построенном на В - дереве. Тем самым мы получаем лексикографически отсортированный словарь. В процессе выполнения работы возникла сложность с удалением элемента из дерева, т.к. для этого приходилось писать дополнительные функции проверки и перестройки дерева, а также с ускорением.