Лабораторная работа №2 по курсу дискретного анализа: сбалансированные деревья.

Выполнил студент группы М8О-208Б-20 МАИ Примаченко Александр.

Условие

Вариант алгоритма: 0 Декартово дерево

Реализовать декартово дерево с возможностью поиска, добавления и удаления элементов.
Необходимо создать программную библиотеку, реализующую указанную структуру данных, на основе которой разработать программу-словарь. В словаре каждому ключу, представляющему из себя регистр независимую 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 файла до его окончания. Каждая строка может иметь следующий формат:
Программа должна вывести строку «OK», если операция прошла успешно, «Exist», если слово уже находится в словаре.
Программа должна вывести «OK», если слово существовало и было удалено, «NoSuchWord», если слово в словаре не было найдено.
Программа должна вывести «OK: 34», если слово было найдено; число, которое следует за «OK:» — номер, присвоенный слову при добавлении. В случае, если слово в словаре не было обнаружено, нужно вывести строку «NoSuchWord».

Метод решения

Декартово дерево – это структура данных, которая объединяет в себя: бинарные дерево поиска и бинарную пирамиду (кучи). Состоит оно из пары ключа и приоритета.
Основными операциями декартово дерева является поиск, вставка и удаления.

1. Операция поиск. Происходит точно также, как и в бинарном дерево поиска.
2. Операция вставка. Сначала вставка осуществляется, как БДП и сравниваем по ключам. При вставке нарушается порядок приоритета. Чтобы исправить это нужно вставляемый элемент сравнивать его приоритет с приоритетом предка. Если приоритет меньше, то делается поворот (левый/правый зависит какой сын). Условаие остановки: либо

стал элемент корнем, либо приоритет найден больше, чем у вставляемого элемента.
3. Операция удаления. Если элемент лист, то просто удаляем значения. Если удалить не лист с одним или двумя потомками. Один потомок делается поворот относительно данного элемента и его потомка. Два потомка выбирается из двух у кого больший приоритет и делается поворот относительно данного элемента и элемента потомка с большим приоритетом. Далее пытаемся сохранить свойства БДП и пирамиды. Потом идем до конца, пока удаляемый элемент не станет листом и его можно легко удалить.

Еще один способ для реализации основных операций декартовых деревьем является операций merge и split. Данный операции является операциям объединение и разбиение. Этим способом были и написаны операции в данной лабораторной работе.

Описание программы

В данной программе содержится один файл, где реализованы операции поиск, вставка и удаление, а также вспомогательные операции merge и split.

Дневник отладки

Ошибка не выявлены.

Тест производительности

Число операций: 1000, 10000, 1000000.
1. Декартово дерево: 0.146s, 0.289s, 2.791s.
2. std::map: 0.148s, 0.609s, 6.677s.

Откуда можно заметить, что декартово дерево работает быстрее нежели стандартный std::map.

Тест производительности

В данной лабораторной работе было предложено изучить некоторые виды алгоритмов сбалансированных деревьев. Мной был реализован алгоритм декартово дерево.
Данный алгоритм является достаточно быстрым и выполняется за O(h), где h-высота дерева. Также было изучены дополнительные операции merge и split, которые помогают реализовать операции вставки и удаления.
Я считаю лабораторная работа оказалась достаточно полезной. Ведь сбалансированные деревья применяется, когда необходимо осуществлять

быстрый поиск элементов, чередующих со вставками новых элементов и удалениями существующих.