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

Выполнил студент группы М8О-207-20 МАИ *Михеева Кристина*.

Условие

1. Реализовать декартово дерево с возможностью поиска, добавления и удаления элементов. Тип ключей и значений, а также формат входных и выходных данных такой же, как в обычных вариантах, команд !Save и !Load в тестах нет.

   Программа должна обрабатывать строки входного файла до его окончания. Каждая строка может иметь следующий формат: + word 34 — добавить слово «word» с номером 34 в словарь. Программа должна вывести строку «OK», если операция прошла успешно, «Exist», если слово уже находится в словаре. - word — удалить слово «word» из словаря. Программа должна вывести «OK», если слово существовало и было удалено, «NoSuchWord», если слово в словаре не было найдено. word — найти в словаре слово «word». 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».

2. Вариант алгоритма: Декартово дерево.

Метод решения

Декартово дерево - это структура данных, которая объединяет в себя: бинарное дерево поиска и бинарную пирамиду (кучи). Состоит оно из пары ключа и приоритета.

Основными операциями декартово дерева являются поиск, вставка и удаление.

1) Операция поиск. Происходит точно также как и в бинарном дереве поиска.

2) Операция вставка. Сначала вставка осуществляется как БДП и сравниваем по ключам. При вставке нарушается порядок приоритета. Чтобы испрвать это нужно вставляемый элемент сравнивать его приоритет с приоритетом предка. Если приоритет меньше, то делается поворот(левый/правый зависит какой сын). Условие остановки: либо стал элемент корнем, либо приоритет найден больше, чем у вставялемого 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 одним способ для реализации основных опреций декартовых деревьем являются оперции merge и split. Данный операции являются операциям объеднение 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 вспомогаельные операции merge и split.

Дневник отладки

Ошибки не выявлены.

Тест производительности

Число операций: 1000, 10000, 1000000.

Декартово дерево: 0.146s, 0.289s, 2.791s.

std::map. : 0.148s, 0.609s, 6.677ms.

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

Выводы

В данной лабораторной работе было предложено изучить некоторые виды алгоритмов сбалансированных деревьев. Мной был реализован алгоритм декартово дерево.

Данный алгоритм является достаточно быстрым выполняется за O(h), где h-высота дерева. Также было изучены дополнительные операции merge и split, которые помогают реализовать операцию вставка и удаления.

Я считаю лабораторная работа оказалась доастаточно полезной. Ведь сбалансированные деревья применяется, когда необходимо осуществлять быстрый поиск элементов, чередующийся со вставками новых элементов и удалениями существующих.