

Лабораторная работа № 2 по курсу дискретного анализа: реализация словаря на основе дерева

Выполнил студент группы 08-207 МАИ *Днепров Иван*.

Условие

Необходимо создать программную библиотеку, реализующую указанную структуру данных, на основе которой разработать программу-словарь:

1. 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 264 - 1. Разным словам может быть поставлен в соответствие один и тот же номер.
2. 3. PATRICIA.

Метод решения

Программа читает стандартный поток и парсит его. В зависимости от входных данных вызываются различные функции дерева, либо выводится сообщение об ошибке, если не получилось соотнести строку с командой.

Описание программы

Считывание текста и парсинг текста реализованы функции ReadLine. Вызов ReadLine и функция дерева происходит из функции main. Сериализация дерева также является функцией структуры PATRICIA, как и функции добавления, поиска и удаления элемента дерева. Сериализация PATRICIA реализована рекурсивно, она записывает бинарный файл корень дерева, затем добавляет всех его детей, добавляет обозначение того, что дети закончились, за тем то же самое делает для братьев. Смысл в том, что благодаря тому, что при выгрузке дерева мы сохраняем его структуру, нам не надо думать, как построить дерево, мы просто зачитываем данные и добавляем либо сына, либо рата, либо поднимаемся на уровень выше.

Дневник отладки

В начале я не верно строил дерево, благодаря чему терялось часть элементов. Потом я забыл про регистронезависимость и не добавил это, в итоге у меня есть два типа функций: регистронезависимые и регистрозависимые. За тем я понял, что не предусмотрел случай добавления или удаления элемента, при котором надо разбивать или сращивать ноду. Программа не проходила по времени за счет того, что я не добавил оптимизирующие ключи компиляции.

Выводы

Словарь на основе дерева PATRICIA очень эффективен для обработки и хранения больших объемов длинных данных, например для хранения и поиска строковых данных большой длины. Я использовал PATRICIA в четвертой лабораторной, чтобы повысить скорость поиска элемента в словаре, для тестов с небольшим размером словаря – это не самое эффективное решение, но для больших словарей оно значительно эффективнее линейного списка, который я использовал до этого. За частую PATRICIA применяется для построения словарей для задач, в которых критичны как скорость, так и компактность хранения данных. По сравнению с простым trie. PATRICIA гораздо компактнее за счет того, что в ноде этого дерева хранятся все несовпадающие элементы, в то время как в trie каждый элемент хранится в отдельной ноде. Такой способ хранения не только позволяет более компактно хранить данные, но и обеспечивает более высокую скорость поиска слова. Но за счет того, добавление и удаление элементов в PATRICIA гораздо сложнее, чем в классическом trie, эти функции работают медленнее.