

Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №2 по курсу «Дискретный анализ»

Студент: К. А. Калугин
Преподаватель: А. А. Кухтичев
Группа: М8О-207Б
Дата:
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №2

Задача: 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 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».

! Save /path/to/file — сохранить словарь в бинарном компактном представлении на диск в файл, указанный параметром команды. В случае успеха, программа должна вывести «OK», в случае неудачи выполнения операции, программа должна вывести описание ошибки (см. ниже).

! Load /path/to/file — загрузить словарь из файла. Предполагается, что файл был ранее подготовлен при помощи команды Save. В случае успеха, программа должна вывести строку «OK», а загруженный словарь должен заменить текущий (с которым происходит работа); в случае неуспеха, должна быть выведена диагностика, а рабочий словарь должен остаться без изменений. Кроме системных ошибок, программа должна корректно обрабатывать случаи несовпадения формата указанного файла и представления данных словаря во внешнем файле.

Для всех операций, в случае возникновения системной ошибки (нехватка памяти, отсутствие прав записи и т.п.), программа должна вывести строку, начинающуюся с «ERROR:» и описывающую на английском языке возникшую ошибку.

Вариант дерева: PATRICIA.

Вариант ключа: Последовательность букв английского алфавита длиной не более 256 символов.

Вариант значения: Числа от 0 до $2^{64} - 1$.

1 Описание

Patricia чем-то напоминает обычное бинарное дерево, однако обладает определенными особенностями. Во-первых, в ней существуют ссылки вверх, причем на каждый узел указывает одна обратная ссылка. Во-вторых, в каждом узле хранятся дополнительные данные - номер биты для проверки. Эти номера возрастают по мере продвижения вниз по дереву. В-третьих, в моей реализации используется фиктивный корень. У него есть лишь одна ссылка - левая, а номер бита так же является фиктивным - по сути, это -1.

2 Исходный код

Пользователь вводит команды заданного вида. Они считываются в качестве строк и, в зависимости от команды, запускается соответствующая функция - поиск, добавление, удаление, сохранение или загрузка. Добавление выполняет проверку отсутствия в дереве требуемого слова, после чего добавляет его. Удаление, соответственно, после проверки удаляет слово. Сохранение и загрузка открывают файлы, после чего записывают или считывают данные в определенном формате. Так же, попутно подсчитывается контрольная сумма - в случае несоответствия ожидаемого формата и реального выводится ошибка, а загруженное дерево удаляется.

2.cpp	
void Clear ()	Функция очистки строки
int code (char c)	Функция получения кода буквы
bool bitcheck (int number, TString & str)	Функция проверки бита слова
TNode* ParFind (TString& word, TNode** prev, const int& difbit, bool& f)	Функция поиска 1
TNode* FastParFind (TString& word, TNode** prev, bool& f, TNode* current)	Функция поиска 2
TNode* FastExtraParFind (TNode* nodeWord, TNode** prev, bool& f, TNode* current, TNode* & trueparent)	Функция поиска 3
TNode* Find (TString& word)	Функция поиска 4
void Add (TKV& addingnode)	Функция добавления элемента
void Delete (TString & deletingnodekey)	Функция удаления элемента
void Clean (TNode* current)	Функция очистки дерева
void RecParcer (TNode* current, FILE* fd, unsigned long long int& corres, unsigned long long int& res)	Вспомогательная функция для сохранения
void Save (TString & filename)	Функция сохранения
TNode* RecReader (FILE* fd, TNode** vault, unsigned long long int& corres, unsigned long long int& res)	Вспомогательная функция для загрузки
void Load (TString & filename)	Функция загрузки

3 Консоль

```
PS C:>.\a.exe
+ a 1
OK
+ A 2
Exist
+ aa 18446744073709551615
OK
aa
OK: 18446744073709551615
A
OK: 1
-A
OK
a
NoSuchWord
```

4 Тест производительности

Сравним время выполнения теста на 10^5 и $5 * 10^5$ команд. Для сравнения возьмем стандартную структуру map.

```
PS C:\VSC\DA>Get-Content .\test | .\patricia.exe
627 ms
PS C:\VSC\DA>Get-Content .\test2 | .\patricia.exe
3726 ms
PS C:\VSC\DA>Get-Content .\test | .\map.exe
2652 ms
PS C:\VSC\DA>Get-Content .\test2 | .\map.exe
20067 ms
```

Как видно, Patricia работает быстрее, чем map. Это достигается за счет того, что в достаточно больших тестах встречаются строки, которые начинаются с одинаковых подстрок. Именно на работу с такими строками и рассчитана patricia, что ускоряет ее работу.

5 Выводы

Выполнив ЛР №2 я научился:

- 1) Представлять данные в компактном и удобном для записи формате.
- 2) Активно комментировать особо сложные участки кода
- 3) Пользоваться ссылками для передачи параметров в функции.

Список литературы

- [1] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн.
Алгоритмы: построение и анализ, 2-е издание. — Издательский дом «Вильямс»,
2007. Перевод с английского: И. В. Красиков, Н. А. Орехова, В. Н. Романов. —
1296 с. (ISBN 5-8459-0857-4 (рус.))
- [2] *Дерево Патриция и с чем её едят?*
URL: <https://medium.com/@grislava/дерево-патриция-и-с-чем-её-едят-a85c9f454bc4>.