Dana Vrajitoru
C243 Data Structures

Homework 5

Due date: Tuesday, October 14, 2014.

Ex. 1. a. Download the following files:
WordCounter.h
WordCounter.cc
HashTable.h
HashTable.cc
porter.h
porter.cc
main.cc
stopWords
Makefile

You can also download the following two files with examples of using the vector and list classes from the STL:
list_test.cc
vector_test.cc

Store the files in a special folder and compile them with the command "make". The executable created by the command make is called "hashing".

Program description

The program is reading in a list of words terminated by an enter followed by Ctrl-D (end of file) for Linux, or Ctrl-C on Windows. The program is performing an indexing of these words using two hash tables.

The program first reads in a list of common words from the file stopWords and stores them in a hash table called stopList. Note that if you run this on Windows, the program might have a hard time finding the path to the file "stopWords".

In the second part, the program inputs from the console the list of words entered by the user. For each of them, it calls the functions toLowerCase and then clean to convert the word to lowercase and remove all the spaces and punctuation from it.

At this point the program checks if the words are in the list of common words (the stopList). If they are, they will simply be discarded. If they are not, then they are processed by the Porter transformation (function stripAffixes) which removes prefixes and suffixes to keep only the root of the words.

A second table will keep count of the frequency of each particular root in the text entered by the user. Thus, it will store the root and the number of times it has encountered it (from the same word or from different words). At the end of the input text, the program displays a statistic of word frequencies.

The program is not functioning yet for two reasons. Some functions in the class HashTable have not yet been

implemented, and some code needs to be added into the main program.

b. Complete the implementation of the class HashTable with the 6 functions that don't have a proper implementation yet. Inside the body of these functions you'll find a comment saying that the code must be supplied by the student. Read the comments in front of each function carefully for details about their implementation.

For the hashing function, you can choose any function you want. I suggest selecting one of the functions recommended as "good" by the textbook (chapter 8), or the notes (chapter 11).

c. Complete the function indexWords according to the comment starting with "For the student to add". Implement the function increment in the same file.

d. (* 2 extra credit points) Make some experiences with 3 different hashing functions and choose which one is the best. Write a small paragraph to report the results of your experiences and to explain your reasons why you think the function you chose is the best one.