Darren Fahy
COSC 160
5/1/17

Spell Checker Writeup

I wrote this program in C++ and it takes in 2 or 3 command line arguments. If 2 are supplied it knows the first is the dictionary and the second is the text file. If three are supplied the third argument is the personal dictionary.

I have a function to count the number of entries in the dictionary and the personal dictionary files. It just opens the files and counts the lines.

I created another function to pick a table size. I didn't really want to hard code a table size. I made a little array with some possible prime table sizes ranging from 83 to 514,903. The table size function just finds a prime in the list greater than 2 times the number of dictionary entries. Please don't use a list greater than 257,000 words. Wasn't necessary but I didn't want to hard code a size.

My hash map class has a constructor that takes in an int as a parameter—the table size. I use insert wrapper functions and hash the object using Horner's rule where I raise each char in a string's ascii value times a prime number (23) to the position of the char in the string. The function is templatized to take any T object. I take the absolute value of that number (sometimes it was negative if it got too big) and mod it by the table size to create the hash key.

I used chaining to handle collisions. When the array pointed to a node that was not null, the insert function used previous and current pointers to node objects to connect a new node. I used counters to keep track of collisions along the way. Created a new function to calculate the average chain length. It iterates through the list and counts the number of chains and chain length. Might not be the most efficient way but my program ran pretty quickly so I didn't need to mess with it.

Finally, I output the relevant information to a log file. You can change the logPath string in main. Used ios::app to append the file. Calculated lambda by dividing total number of dictionary entries by the size of the table. It's roughly .5 because I multiplied the list size by 2 to find a prime table size.