CS 284D: Homework Assignment 4
Due: May 12nd, 11:55pm

1 Assignment Policies

Collaboration and IP Policy. Do homework individually. Do ask questions if anything is not clear. You can collaborate to understand the material. Do not solve the problem with others. Do not use any solution on the Internet. Do not exchange code with other students. **Any solution found online or in another submission will be penalized.**

Late Policy. No late submissions will be allowed without consent from the instructor. For urgent or unusual circumstances e-mail the instructor.

2 Assignment: Hashing

The goal of this assignment is to test three hashing implementations. You will also use your best hashing implementation for a simple spell-checker. Test each hashing implementation as follows:

1. Read all words from a given text file words.txt and insert them into a hash table. After the table is created print out the total number of elements in the table (N), the size of table (T), the load factor (N/T), the total number of collisions (C), and the average number of collisions (C/N).
2. Check whether each word in another given file query_words.txt is in the hash table or not. For each word that is found print the word, the string "Found" and the number of probes used. For each word not found, print the word, the string "Not Found" and the number of probes used.

Instructions To implement the above, write a test program named "CreateAndTestHash". **Choose one** of the following two ways to input the arguments.

1. Your programs should run from the terminal as follows:

   "CreateAndTestHash <words file name> <query words file name> <flag>"

   <flag> should be quadratic for quadratic probing, linear for linear probing, and double for double hashing.

For example you can write on the terminal:

“CreateAndTestHash words.txt query_words.txt quadratic”

2. Your program should run and prompt the user for three input arguments. These include (1) the name of the file containing the text <words file name>, (2) the name of the file containing the query words <query words file name>, (3) and the type of probing/hashing to be done with the Scanner class <flag>. <flag> should be quadratic for quadratic probing, linear for linear probing, and double for double hashing.

   For example, after you start the program “CreateAndTestHash”, the user will need to input “words.txt”, then input “query_words.txt”, and finally input “quadratic”.

2.1 Part1 (20 points)

Modify the code for quadratic and linear probing and test CreateAndTestHash.

2.2 Part 2 (30 points)

Write code to implement double hashing and test using CreateAndTestHash. This is going to be a variation of quadratic probing. The difference will lie in function FindPos() that has to now provide probes using a different strategy. As the second hash function you can try: $R - (x \mod R)$ where R is a prime smaller than table size. Try various values for R and keep the one that provides the best performance (i.e. smaller number of collisions). In your Readme file state the exact R you are using. Note that again you should provide templated testing functions so that testing code is not replicated.

2.3 Part 3 (50 points)

Now you are ready to implement a spell checker by using your favorite hash table. Given a document your program should output all of the misspelled words. For each misspelled word you should also provide a list of candidate corrections from the dictionary that can be formed by applying one of the following rules to the misspelled word:

1. Adding one character in each possible position
2. Removing one character from the word
3. Swapping adjacent characters in the word

Your program should run as follows:
“SpellCheck <document file> <dictionary file>”
You will be provided with a small document named document1_short.txt (and document1.txt) and a dictionary file with approximately 100k words named wordsEn.txt.

Alternatively, your program can prompt the user for the name of the document file and the dictionary file.

As an example, the following mistakes should be corrected by your spell checker:

- “comlete -> complete (case a)”

- “deciasion -> decision (case b)”
- “lwa -> law (case c)”

Note that there may be other words in the document that are not in the dictionary due to the limited size and scope of the dictionary. You do not have to correct those misspellings.

Submission Instructions:

Please submit a zip archive named hw5.zip with a spell check and related files in it.

Please make sure your name and pledge are on all documents. No credit will be given for submissions with other file types.

IP Policy By receiving this homework as part of me taking the class CS284D I acknowledge that the content of this document will not be circulated as a whole or in part and in any implicit or explicit way. Anything else breaches the code of student conduct and, importantly, violates the federal laws of the US for intellectual property (cf. U.S. Copyright Law, December 2016). The instructor of this course maintains in its entirety the copyright of every document or electronic material (including slides).