BLG 335E Analysis of Algorithms I

Project 3

Tuesday 27th November, 2018

Due: 16th Dec, 2018

In this project, you are asked to implement several hash functions on the given input files to store dataset in to a m-sized hash table. There are two input files: "vocab.txt" and "search.txt". First file contains unique tokens which are collected from an English corpus. Each token exists in a line and index values of tokens are corresponding to in which line they are. The other file is given to be used in searching operations. This file contains tokens which are searched in a hash table after hash tables are constituted according to different hash functions.

1. (15 points) Implement a m-sized hash table that uses linear probing strategy in order to store string elements by using the following hash function. Each slot of your hash table can only contain a string variable. (Note: You do not store *i* values during the insertion operation. You will use *i* values in searching operation as how you do in insertion operation.)

$$h(k,i) = (k+i)\ mod\ m \text{ and } i \in [0, m-1]$$

2. (15 points) Implement a m-sized hash table that uses double hashing strategy in order to store string elements by using the following hash function. Each slot of your hash table can only contain a string variable.

$$h(k, i) = (h_1(k) + i * h_2(k)) \bmod m \text{ and } i \in [0, m-1]$$

$$h_1(k) = k \bmod m$$

$$h_2(k) = p - (k \bmod p),\ p \text{ is a prime number and } p \in [0, m-1]$$

3. **(20 points)** Implement a m-sized hash table that uses universal hashing strategy in order to store string elements. In the case of any collision, you can get help from any open addressing strategy (e.g. as in linear probing). Details of the universal hashing strategy is given below.

Step #1: Choose the table size m to be prime.

Step #2: Decompose the key *k* into 2 digits (at the end, *r* chunks) so that $k = \langle k_0, k_1, \ldots, k_r \rangle$ and $0 \le k_i < m$. For this homework, r=3.

Step #3: Pick $a = [a_0, a_1, \ldots, a_r]$ where each a_i is generated randomly from $[0, m-1]$ interval.

Step #4: Apply hashing function given below:

$$h_a(k) = \sum_{i=0}^{r} a_i k_i \mod m$$

Example for decomposition for *k*:

k= 96356→k_0=9, k_1= 63, k_2=56
k= 1398→k_0=0, k_1= 13, k_2= 98
k= 548→k_0=0, k_1= 5, k_2= 48
k= 83→k_0=0, k_1= 0, k_2= 83
k= 7→k_0=0, k_1= 0, k_2= 7

4. **(30 points)** Search Functions
Write three functions and each of them searchs given set of keys in the hash table that you implemented in (1), (2) and (3) respectively.

5. (20 points) The situation where a newly inserted key maps to an already occupied slot in hash table is called **collision.** In this part, hashing funtions which are implemented in this homework are compared according to collision numbers occur during the insertion / search operations.

Number of collisions when inserting/searching an item *i* is defined as follows:

```
collisions = 0
    Compute h(k,i)
    if  h(k,i) is not empty then
        increment collisions by 1.
```

Using the definition above, insert all the elements in "vocab.txt" to the hash tables, compute the number of collisions for each strategy and fill in the table with the number of collisions occured for each strategy and m value. Similarly, search all the elements in "search.txt" and compute the number of collisions for each strategy. Comment on the results.

	Insertion		
	Linear	Double	Universal
m = 17863			
m = 21929			

	Search		
	Linear	Double	Universal
m = 17863			
m = 21929			

DETAILED INSTRUCTIONS:

- All your code must be written in C++ using object oriented approach and able to compile and run on Linux using g++.
- Your code should run with this arguments via terminal:

  g++ <your_cpp_file.cpp> -o output.o
  ./output.o vocab.txt search.txt
- Submissions will be done through the Ninova server. You must submit all your program and header files. You must also submit a softcopy report.

- Each student must work individually for the project. This is not a group assignment and getting involved in any kind of cheating is subject to disciplinary actions. Your homework SHOULD NOT include any copy-paste material (from the Internet or from someone else's paper/thesis/project).
- Submissions are made through the Ninova system and have a strict deadline. Assignments submitted after the deadline will **NOT** be accepted. If you send your homework via e-mail, you will NOT get any points. Don't wait until the last minute. Upload whatever you have, you can always overwrite it afterwards.
- If you have any questions, please feel free to contact Res. Asst. Tuğba PAMAY via e-mail (pamay@itu.edu.tr).