Spring 2023 – CS 303 Algorithms and Data Structures
Lab 7

Objectives:

- Implement a hashing function.
- Use the hashing function to create a hashmap implementation.

Problems

1. Create a hashmap that is made of elements HashElement(int Key, String Value). The size of hashmap will be 100. Implement the following methods for the hashmap:

    a. put(int Key, String Value): Puts the key value pair in the hashmap at a certain index. You need to implement a simple hash function H(Key)=Key mod mapsize to find the index where you will put the pair. If collision occurs, i.e., a pair already exists in that index and the key is not the same as the current key, then you will use this function to resolve the collision, H(x)=(7*H(x)+1) mod mapsize, until you get an empty slot. If the index is already full and the keys are the same, just replace the old value with the new one.

    b. get(int Key): Gets the value associated with the Key. You should not do linear search throughout the hashmap for the key, rather you will calculate the index using the hash function stated above, go directly to that index and retrieve the value.

2. Write a driver program to test your implementation of hashmap. Allow the user to put or get data.

3. Implement linear probing to put a new value in your HashMap. The sequence of probes are: H(x) mod mapsize, (H(x) +1) mod mapsize, (H(x)+2) mod mapsize, (H(x)+3) mod mapsize … and so on.

4. Implement quadratic probing to put a new value in your HashMap. The sequence of probes are: H(x) mod mapsize, (H(x) +1) mod mapsize, (H(x)+4) mod mapsize, (H(x)+9) mod mapsize … and so on.

5. Use the hashmap implementation from the previous question (make sure to update the hashmap size) to store the given input file that consists of two fields: a UPC key and the corresponding description. Use the hashmap created to find the description associated with a given set of UPC keys. The input file UPC.csv provides the key and corresponding descriptions in a comma separated file and the various search keys are provided in the file input.dat. First test the program by entering couple of keys manually and print the description. Once you are convinced the program is working correctly, test the program for the given search keys and determine the total time taken to complete the search.

Compare the times for searching the keys using the given function, linear probing, quadratic probing with hashmap

Submission:

- You are required to submit a written report (.pdf), ICF, and your project files turned in separately on Canvas, no .zip files
- Homework report must follow the guidelines provided in the sample report uploaded in Canvas. Please include the screenshot of your code and outputs of your code at the end of your report.
- Do not forget to submit Independent Completion Form
- When you create the code to read the file use relative path instead of absolute path

DATA

HashEntry.java	HashEntry.py
HashMap.java	HashMap.py
input.dat	UPC-random.csv
UPC.csv	

Grading Rubric

Coding	Implementing Algorithms	20 points
	Producing Correct Outputs	20 points
Report	Explaining the algorithms used	10 points
	Displaying the output with a graph or table	20 points
	Comparing the outputs and discussing the time complexity of algorithms	20 points
	Correct submissions of the files (ICF, Code.zip, report.pdf)	10 points