CSC383 Programming Project
Developing an Extendable Hashing Simulator

College of Computer and Information Sciences
King Saud University

SPRING 2019

1 Introduction

Dynamic hashing techniques allow the hash function to be modified dynamically to accommodate the growth or shrinkage of the database. Obviously, dynamic hashing overcomes static hashing problems where the number of buckets is fixed and does not change. A well-known technique of dynamic hashing is extendable hashing which copes with changes in database size by splitting and coalescing buckets as the database grows and shrinks. As a result, space efficiency is retained. Moreover, since the reorganization is performed on only one bucket at a time, the resulting performance overhead is acceptably low.

Your goal in this project is to implement an extendable hashing simulator with three basic operations: lookup, insert and delete. Your simulator should be linked with a main program in-order to test its functionality. Note: you should choose the appropriate data structure for the software.

2 Requirements

In this project you are required to implement an extendable hashing simulator to fulfill the following requirements:

1. Read the bucket size from the user.
2. Read a search key from the user in order to apply one of the available operations.
   - The search key is a positive integer k such that $0 \leq k \leq 255$
   - Use the given hash function (see appendix in Section 4) to calculate an $8-$bit Hash value
   - Use the first i high order bits of the hash value as a displacement into bucket address, where i is the directory global depth
3. There are three main operations can be performed by the user: (1) lookup a record by its key, (2) insert a record, (3) delete a record.

4. The simulator should be linked to a main test program. Pay attention to that, it should visualize the hash directory and buckets in an efficient way similar to the one used in book's examples.
    - Use Graphical User Interface (GUI) tools to achieve this goal.

3 Deliverable and Rules

The submission deadline is 28/03/2019, you should deliver:

1. A report written using the provided template.
2. Source code submission to LMS.

You have to read and follow the following rules:

1. This assignment is to be conducted by groups of exactly three students.
2. The students can implement their own data structures; use Java collections, or any other library to achieve assignment's requirements.
3. Every group member should participate in all parts of the assignment: designing, programming, and writing the software and report. Members of the same group may receive different marks according to their participation in the work.
4. The submitted software will be evaluated in a demonstration which all the group members should attend. In the demonstration you will be given problems to solve in a real time.
5. Any member of the group who fails to attend the demonstration without a proper excuse shall receive 0 in the demo mark.
6. In accordance with the university regulation, cheating in the project will be penalized by the mark 0 in the project.

4 Appendix

1. Hash Function Source Code

```
// Input: A positive integer key such that 0 <= key <= 255
// Output: A string contains an 8-bit hash value of the received key

public static String hash(int key) {
   String value = Integer.toBinaryString(key);
   for (int i = value.length(); i < 8; i++)
      value += "0";
   return value;
}
```