CSCI 3104 Spring 2022
Instructors: Profs. Chen and Layer

Quiz 24 - Hash Tables

Due Date ..... April 22
Name ..... **Alex Ojemann**
Student ID **109722375**

Contents

1 Instructions

- The solutions **should be typed**, using proper mathematical notation. We cannot accept hand-written solutions. Here's a short intro to LaTeX.
- You should submit your work through the **class Canvas page** only. Please submit one PDF file, compiled using this LaTeX template.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please submit this document with no fewer pages than the blank template (or Gradescope has issues with it).
- You **may not collaborate with other students**. **Copying from any source is an Honor Code violation. Furthermore, all submissions must be in your own words and reflect your understanding of the material.** If there is any confusion about this policy, it is your responsibility to clarify before the due date.
- Posting to **any** service including, but not limited to Chegg, Discord, Reddit, StackExchange, etc., for help on an assignment is a violation of the Honor Code.

2 Standard 24 - Hash Tables

Problem 1. Consider the hash function $h(k) = k \ mod \ 2$ for all keys k for a table of size 2. (Resolve collisions by chaining with a linked list.) You have three applications:

- Application 1: keys are randomly drawn from *even* integers
- Application 2: keys are randomly drawn from *odd* integers
- Application 3: keys are randomly drawn from *all* integers

1. *For which application does the hash function $h(k)$ perform better? Please* **explain/justify adequately** *your answer.*

   Answer. h(k) performs best for Application 3, the case where keys are drawn from all integers. This is because when keys can be either even or odd, the keys are equally likely to be placed into each of the buckets. Thus the load factor is n/2 because n keys are equally likely to be placed into 2 buckets. However, for the cases in which all keys are only even or only odd, the keys will only be placed into one of the buckets which results in a load factor of n because n keys are all placed in 1 bucket. The performance of a hash table's lookup and delete operations is inversely related to the load factor, so application 3 has the best performance for these operations. □

2. *In each of the three applications, does the hash function $h(k)$ satisfy the uniform hashing property? Please* **explain/justify adequately** *your answer.*

   Answer. The uniform hashing property is satisfied if and only if the keys are equally likely to hash into each of the buckets. Applications 1 and 2 don't satisfy this because the keys are all assigned to one of the two buckets. Application 3 does satisfy this because the keys are randomly drawn and the potential keys include all integers so they're equally likely to be odd or even thus equally likely to be hashed into each bucket. □