CSCI 3104 Fall 2021
Instructors: Profs. Grochow and Waggoner

Midterm 2- Standard 25

Due Date ..... Nov / 18^{th} / 2021
Name ..... **Michael Ghattas**
Student ID **109200649**

Contents

1 Instructions **1**

2 Honor Code (Make Sure to Virtually Sign) **2**

3 Standard 25- Hashing and Collisions **3**

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.
- You **must** virtually sign the Honor Code (see Section 2). Failure to do so will result in your assignment not being graded.

2 Honor Code (Make Sure to Virtually Sign)

Problem 1.

- My submission is in my own words and reflects my understanding of the material.
- I have not collaborated with any other person.
- I have not posted to external services including, but not limited to Chegg, Discord, Reddit, StackExchange, etc.
- I have neither copied nor provided others solutions they can copy.

I agree to the above, Michael Ghattas. □

3 Standard 25- Hashing and Collisions

Problem 2. Consider a hash table designed to store integers, using the hash function $h(k) = k \ mod \ 3$ for all keys k for a table of size 3. (Resolve collisions by chaining with a linked list.) You have three scenarios:

- Scenario 1: keys are randomly drawn from integers that are divisible by 3.
- Scenario 2: keys are randomly drawn from integers of the form $3m + 1$ where m is an integer.
- Scenario 3: keys are randomly drawn from *all* integers

Do the following.

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

Answer:

- **Scenario 1:** If keys are randomly drawn from integers that are divisible by 3, then we are only using index (0) in the hash-table. Thus we have $\Theta(1 + \frac{n}{1}) = \Theta(1 + n)$
- **Scenario 2:** If keys are randomly drawn from integers of the form 3m + 1, where m is an integer, then we are only using index (1) in the hash-table. Thus we have $\Theta(1 + \frac{n}{1}) = \Theta(1 + n)$
- **Scenario 3:** If keys are randomly drawn from all integers, then we are using all three indices (0, 1, 2) in the hash-table. Thus we have $\Theta(1 + \frac{n}{3})$

$(1 + \frac{n}{3}) \in O(1 + n) \rightarrow$ **Thus scenario 3 performs best since it has the smallest** $load \ factor$**.** □

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

Answer:

Yes, as it satisfies the the following two conditions of *Simple Uniform Hashing Assumptions*:

- For any key k and any index $i \in \{0, 1, 2\}$ $Pr[h(k) = i] = 1/3$. That is, the probability that h(x) maps the element k into position i is 1/m.
- For any two keys k_1, k_2 and any index i, $Pr[h(k_1) = i]$ and $Pr[h(k_2) = i]$ are independent. As for any key k $\{Pr[h(k) = i] = \frac{1}{3}\}$ independently.

□

(c) Suppose you have n keys in total for each application. What is the resulting load factor for each application?

Answer:

- **Scenario 1:** Load factor $\alpha = \frac{n}{1} = n$
- **Scenario 2:** Load factor $\alpha = \frac{n}{1} = n$
- **Scenario 3:** Load factor $\alpha = \frac{n}{3}$

□

(d) Suppose you have n keys in total for each application. What are the time complexities of the dictionary operations: add, delete, and find, respectively?

Answer:

Scenario 1: Add $\Theta(1)$, Delete $\Theta(n)$, and Find $\Theta(n)$.

Scenario 2: Add $\Theta(1)$, Delete $\Theta(n)$, and Find $\Theta(n)$.

Scenario 3: Add $\Theta(1)$, Delete $\Theta(\frac{n}{3})$, and Find $\Theta(\frac{n}{3})$.

□