CSCI 3104 Fall 2021
Instructors: Profs. Grochow and Waggoner

Problem Set 10

Due Date .... Nov / 16^{th} / 2021
Name .... **Michael Ghattas**
Student ID **109200649**
Collaborators .... **Me, Myself & I**

Contents

1 Instructions

- The solutions **must 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 are welcome and encouraged to collaborate with your classmates, as well as consult outside resources. You must **cite your sources in this document. 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, 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. $\square$

3 Standard 25- Hash Tables

3.1 Problem 2

Problem 2. Hash tables and balanced binary trees can be both be used to implement a dictionary data structure, which supports insertion, deletion, and lookup operations. In balanced binary trees containing n elements, the runtime of all operations is $\Theta(\log n)$.

For each of the following three scenarios, compare the average-case performance of a dictionary implemented with a hash table (which resolves collisions with chaining using doubly-linked lists) to a dictionary implemented with a balanced binary tree. That is, for each of the three scenarios: describe clearly in terms of big-Oh notation the average-case runtime of the dictionary implemented with a hash table, and then clearly state whether the hash table implementation is better or the balance binary tree implementation is better, and why.

(a) A hash table with hash function $h_1(x) = 1$ for all keys x.

Answer:

Since we are storing all the keys x in the same constant position, (m is $\Theta(1)$) $\rightarrow \Theta(1 + \frac{n}{m}) = \Theta(1 + \frac{n}{1}) = \Theta(1 + n) = \Theta(n)$, thus $\Theta(n)$ is NOT better than the BBT implementation of $\Theta(\log n)$ as $(\log n) \in O(n)$. $\square$

(b) A hash table with a hash function h_2 that satisfies the Simple Uniform Hashing Assumption, and where the number m of buckets is $\Theta(n)$ (recall that n is the number of items).

Answer:

This means the implementation has a $\Theta(1 + \frac{n}{m}) = \Theta(1 + \alpha)$, thus $\Theta(1 + \alpha)$ is better than the BBT implementation of $\Theta(\log n)$ only when $m > \Omega(\frac{n}{log(n)})$, which in turn causes $(1+\alpha) \in O(\log n)$. The same as BBT implementation of $\Theta(\log n)$ only when $m = \Omega(\frac{n}{log(n)})$, which in turn causes $(1 + \alpha) \in \Theta(\log n)$. And NOT better than the BBT implementation of $\Theta(\log n)$ only when $m < \Omega(\frac{n}{log(n)})$, which in turn causes $(\log n) \in O(1 + \alpha)$. $\square$

(c) A hash table with a hash function h_3 that satisfies the Simple Uniform Hashing Assumption, and where the number m of buckets is $\Theta(\sqrt{n})$.

Answer:

(m is $\Theta(\sqrt{n})$) $\rightarrow \Theta(1+\frac{n}{m}) = \Theta(1+\frac{n}{\sqrt{n}}) = \Theta(1+n^{\frac{1}{2}}) = \Theta(1+\sqrt{n}) = \Theta(\sqrt{n})$, thus $\Theta(\sqrt{n})$ is NOT better than the BBT implementation of $\Theta(\log n)$ as $(\log n) \in O(\sqrt{n})$. $\square$