CSCI 3104 Fall 2021
Instructors: Profs. Chen and Layer

Problem Set 9

Due Date .... April 8
Name .... **Alex Ojemann**
Student ID **109722375**
Collaborators ....

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.

2 Standard 24- Hash Tables

Problem 1. 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.

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

Answer. This hash table will take $\Theta(n)$ time for insertions, lookups and deletions which is worse than a balanced BST. □

(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 $\Theta(n)$.

Answer. This hash table will take $\Theta(1)$ time for insertions, lookups and deletions which is better than a balanced BST. □

(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 $\Theta(n^{3/4})$.

Answer. This hash table will take $\Theta(log(n))$ time for insertions, lookups and deletions which is the same as a balanced BST.

□

3 Standard 25- Doubling Lists and Amortized Analysis

Problem 2. Consider a hash table A that holds data from a fixed, finite universe U. If the hash table starts empty with $b = 10$ buckets and doubles its number of buckets whenever the load factor exceeds $\frac{1}{2}$, what is the load factor after adding $n = 36$ elements to the hash table?

Answer. Start adding elements

Get to 6 elements, load factor $= 3/5 > 1/2$ so set number of buckets to be 20.

Get to 11 elements, load factor $= 11/20 > 1/2$ so set number of buckets to be 40.

Get to 21 elements, load factor $= 21/40 > 1/2$ so set number of buckets to be 80.

At 36 elements in the hash table, the number of buckets is 80. $\square$