Homework 7

ECE 309 Fall 2019
Due: October 16, 2019
Upload an electronic copy of your answers to Moodle under HW7.

*This is a shared google document. This means (1) it may change to clarify content, and (2) other people can view your comments on this file. If you have questions, you are encouraged to comment directly on this document, but **do not add your answers here**. Make a copy into your private Google Drive and then edit the document.*

DO NOT ADD ANSWERS TO THE SHARED DOC! THAT'S CONSIDERED CHEATING!

1. Hash Table

[45 points] Assume we have a hash table for integers that allows duplicates and that a remove operation will remove all copies. Consider the following sequence of operations on a hash table for integers.

- Insert 14
- Insert 714
- Insert 97
- Insert 25
- Insert 6736
- Insert 103
- Insert 3
- Remove 14
- Remove 714
- Search 25

(a) [15 points] Assume the hash table has length 10 and that the hash function is simply index = data % 10. Show a picture of the hash table after executing the operations above assuming Linear Probing is used on collisions. Assume a probeDistance = 5. If there's no place for an item within the probeDistance, then state that, but do not evict or change the size of the table.

0	
1	
2	
3	103
4	
5	
6	25
7	97
8	6736
9	

(b) [15 points] Assume the hash table has length 10 and that the hash function is simply index = data % 10. Show a picture of the hash table after executing the operations above assuming chaining is used.

0	
1	
2	
3	103, 3
4	
5	25
6	6736
7	97
8	
9	

(c) [5 points] For the two hash table designs, which one performs fewer table accesses and comparisons when performing the last search for 25.

The chaining

(d) [10 points] Consider that the size of a hash table array is M entries and the number of items inserted in the set is N. Evaluate the big-O for the amount of memory used for a hash table with linear probing versus a hash table with chaining. State any assumptions in the formation of your answer.

Linear probing – **O(M)** : For worst case, it have to go through entire size of table to search out an element

Chaining – **O(N+M)** : In term of memory, M entries with a list has N elements in.