ICS 311: Algorithms	Spring 2025
Problem Set 6	
Prof. Nodari Sitchinava	*Due: Friday, Feb 21, 2025 at 4pm*

You may discuss the problems with your classmates, however **you must write up the solutions on your own** and **list the names** of every person with whom you discussed each problem.

Start **every** problem on a separate sheet of paper, with the exception of Problems 1 (Peer credit assignment) – Problem 1 can be on the same page as any other problem. Any problem that starts on the same sheet of paper as some other problem will receive 0 points!

1 Peer Credit Assignment (1 point extra credit for replying)

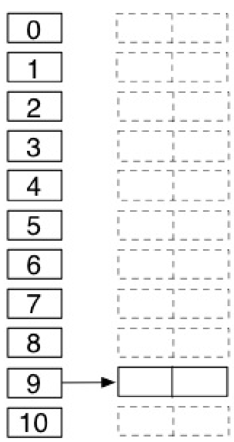
Please list the names of the other members of your peer group for this week and the number of extra credit points you think they deserve for their participation in group work.

- You have a total of 60 points to allocate across all of your peers.
- You can distribute the points equally, give them all to one person, or do something in between.
- You need not allocate all the points available to you.
- ***You cannot allocate any points to yourself!*** Points allocated to yourself will not be recorded.

2 Practice with Hash Tables (24 pts)

Hashing with Chaining

(a) (6 pts) Fill in the table below after inserting 20, 51, 10, 19, 32, 1, 66, 40 (in this order) into an initially empty hash table of size 11 with chaining and $h(k) = k \bmod 11$.



Open Addressing with Linear Probing

(b) (6 pts) Fill in the table below after inserting 20, 51, 10, 19, 32, 1, 66, 40 (in this order) into an initially empty hash table of size 11 with linear probing and

$$h(k, i) \quad = \quad (h'(k) + i) \bmod 11$$

$$h'(k) = k \bmod 11$$

0	1	2	3	4	5	6	7	8	9	10

(c) (3 pts) How many re-hashes after collision are required for this set of keys? *Show your work!*

Open Addressing with Double Hashing

(d) (6 pts) Fill in the table below after inserting 20, 51, 10, 19, 32, 1, 66, 40 (in this order) into an initially empty hash table of size $m = 11$ with double hashing and

$$\begin{aligned} h(k,i) &= (h_1(k) + i \cdot h_2(k)) \bmod 11 \\ h_1(k) &= k \bmod 11 \\ h_2(k) &= 1 + (k \bmod 7) \end{aligned}$$

0	1	2	3	4	5	6	7	8	9	10

(e) (3 pts) How many re-hashes after collision are required for this set of keys? *Show your work!*

3 Expected Number of Collisions in a Hash Table (34 pts)

Suppose we insert n distinct keys into a hash table of size m using independent uniform hashing. What is the expected number of collisions? That is, compute the expected number of pairs of keys $\{k_i, k_j\}$, such that $k_i \neq k_j$ and $h(k_i) = h(k_j)$. Compute the exact value (not just the big-O notation) and show your work. *Hint: Be careful not to double-count the pairs: $\{k_i, k_j\}$ is the same as $\{k_j, k_i\}$.*

4 Hash Table Load Factor (42 pts)

We want to store a set of n keys into a hash table H, that is of size m and uses chaining as the collision resolution method. In this problem you will prove that if the keys are drawn from a universe U of size $|U| > nm$, then regardless of what hash function we use, the worst case runtime for searching in H is $\Theta(n)$.

(a) (7 pts) What is the upper bound for the runtime of searching in H in the worst case? *Use big-O notation and justify your answer.*

(b) (21 pts) Prove that no matter what hash function we use, U always contains a subset of size n consisting of keys that all hash to the same slot. *(Hint: assume that there is no such subset and find a contradiction.)*

(c) (7 pts) What does the claim in part (b) say about the lower bound on the runtime of searching in H in the worst case? *Use big-Ω notation and justify your answer.*

(d) (7 pts) Combine your answers to parts (a) and (c) to conclude that the runtime for searching for a key in H is $\Theta(n)$ in the worst case. How does it compare to searching for an item in a red-black tree, which contains n items from the universe U? When would we want to use a hash table and when would we want to use a red-black tree as a data structure to store and search for items?

5 Hashing (OPTIONAL - 0 pts)

Assume you have an empty hash table of size m with linear probing. What is the probability that the first two keys inserted into this hash table will be placed in adjacent addresses? **Show your work.** *Address 0 and address* $m-1$ *are NOT adjacent.*

6 Analysis of Collisions in Hash Tables (OPTIONAL - 0 pts)

(a) Consider a hash table with m slots that uses chaining for collision resolution. The table is initially empty. What is the probability that, after three keys are inserted, there is a chain of size 3? *Show your work!*

(b) Consider a hash table with m slots that uses open addressing with linear probing. The table is initially empty. A key k_1 is inserted into the table, followed by key k_2. What is the probability that inserting key k_3 requires three probes? *Show your work!*