CIS 1210 — Data Structures and Algorithms
Homework Assignment 8

Assigned: April 8, 2025 **Due:** April 21, 2025

Note: The homework is due **electronically on Gradescope** on April 21, 2025 by 11:59 pm ET. For late submissions, please refer to the Late Submission Policy on the course webpage. You may submit this assignment up to 2 days late.

A. Gradescope: You must select the appropriate pages on Gradescope. Gradescope makes this easy for you: before you submit, it asks you to associate pages with the homework questions. Forgetting to do so will incur a 5% penalty, which cannot be argued against after the fact.

B. LaTeX: You must use the LaTeX template provided on the course website, or a 5% penalty will be incurred. Handwritten solutions or solutions not typeset in LaTeX will not be accepted.

C. Solutions: Please write concise and clear solutions; you will get only a partial credit for correct solutions that are either unnecessarily long or not clear. Please refer to the Written Homework Guidelines for all the requirements. Ed will also contain a complete sample solution in a pinned post.

D. Algorithms: Whenever you present an algorithm, your answer must include 3 separate sections. Please see Ed for an example complete solution.

1. A precise description of your algorithm in English. No pseudocode, no code.
2. Proof of correctness of your algorithm
3. Analysis of the running time complexity of your algorithm

E. Collaboration: You are allowed to discuss **ideas** for solving homework problems in groups of up to 3 people but *you must write your solutions independently*. Also, you must write on your homework the names of the people with whom you discussed. For more on the collaboration policy, please see the course webpage.

F. Outside Resources: Finally, you are not allowed to use *any* material outside of the class notes and the textbook. Any violation of this policy may seriously affect your grade in the class. If you're unsure if something violates our policy, please ask.

NOTE: All final answers must be simplified as much as possible. Each answer should be a single fraction, and you must not involve any combinatorial expressions (e.g., factorial, "choose" notation, etc.). Unsimplified final answers, i.e., final answers involving summations or that are written as sums of multiple terms that could be combined, etc. will lose points.

1. [10 pts] A Wrench in Vedha's Heart

Vedha, the plumber, is known by almost every household–not just for her natural gift to fix any leak or her uncanny ability to tame a water heater–but for her kindness.

She remembers birthdays and always has a gentle smile even when knee-deep in someone else's mess.

That's why National Hug a Plumber Day is so important. To some, it's a meaningless and often forgotten day. But it means something to Vedha. Every year, she looks forward to National Hug a Plumber Day, where she dresses in her nicest overalls and polishes her finest wrench. Sadly, Vedha has grown to not expect hugs on that day. That hope had worn thin over the years, like the soles of her work boots. No one ever remembers.

But, there's still time, and you can help! To help Vedha feel some joy on this year's National Hug a Plumber Day, you need to do the following.

Take the keys: $29, 21, 17, 3, 47, 12, 35, 64, 1$ (in that order) and add them into a hash table of length $m = 12$ using open addressing with the auxiliary hash function $h'(k) = k$. Give the resulting hash table after inserting these keys using:

- **a.** linear probing
- **b.** quadratic probing with $c_1 = 5$ and $c_2 = 4$
- **c.** double hashing with $h_1(k) = k$ and $h_2(k) = 1 + (k \bmod (m - 1))$

2. [15 pts] National Ice Cream Day

For National Ice Cream Day, the famous ice cream connoisseurs Frosty Kevin and Sprinkle Jesse have planned a spectacular giveaway in the whimsical city of *Sunnyland*. They've divided *Sunnyland* into n distinct neighborhoods and they have n ice cream trucks ready to deliver free cones to each neighborhood.

Each ice cream truck independently chooses one of the n neighborhoods uniformly at random—that is, every truck has a $\frac{1}{n}$ chance to serve any given neighborhood. In this festive setup, you can think of each neighborhood as a "slot" in a delivery schedule (similar to a hash table), and each ice cream truck as an "element" that gets assigned to one of these slots.

(a) Let R_i be the random variable that counts the number of ice cream trucks that end up in neighborhood $A[i]$. Give an expression in terms of n and k for $Pr(R_i = k)$ (i.e., the probability that exactly k ice cream trucks deliver cones to neighborhood $A[i]$), where $0 \le k \le n$.

(b) Prove that

$$Pr(R_i = k) \le \left(\frac{3}{k}\right)^k, \quad 0 \le k \le n$$

Note, you may find the inequality $\binom{n}{k} \le \left(\frac{n \cdot e}{k}\right)^k$ useful. Here $e \approx 2.718$ is Euler's number, and you can assume the convention that $\left(\frac{3}{0}\right)^0 = 1$.

3. [20 pts] National Be Kind to Lawyers Day

In honor of National Be Kind to Lawyers Day, the CIS 1210 courts, led by Judge Jake, are digitizing decades of cases to help the overworked attorneys by organizing all the case files.

Each file is represented by a rooted full binary tree. Two files S and T come from the same case if S is a subtree of T or T is a subtree of S. For the purpose of this problem, the trees are stored as pointers to their root node (similar to storing the root node in QuadTree). Each node contains a string value - some message in plain English.

Furthermore, you are given a hash function $h(x)$ which takes in any string x and hashes it into one of 2^{256} unique strings (it always draws from the same set of 2^{256} strings every time). You may assume that each node's string is not too large and can be processed by h in constant time. You may also assume that this hash function obeys SUHA.

Use this powerful hash function to design an algorithm that determines whether two files come from the same case in $O(n_s + n_t)$ time, where n_s, n_t is the number of vertices in S, T respectively. If S is indeed a subtree of T or T is a subtree of S, then your algorithm should always return true. If S is not a subtree of T, and T is not a subtree of S, then your algorithm should return false with very high certainty.

For the proof of correctness, you are not required to provide a full mathematical justification for why the algorithm will return false with very high certainty - a high level explanation will suffice.

Note: A tree S is a subtree of T iff S consists of a node in T and all of this node's descendants, and S is a subgraph of T. For the purposes of this question, two nodes are considered equal if their string messages are the same.

Hint: A string concatenated with the hash of another string can still be used as the input x to our hash function h.

4. [15 pts] Cheesing for Grilled Cheese

As the CIS 1210 TA's designated cheesemonger, Nick has taken it upon himself to lead the annual Grilled Cheese Day celebrations. Determined to give all the TAs a "gouda" time, he's planned a myriad of activities: a "Brie-lliant Bites Tour," a "Curd your Enthusiasm Cheese Making Workshop," and finishing the day with a "Grate Tales Hour" storytime about their favorite cheese-y stories.

To start the festivities off strong, Nick plans to deliver grilled cheese sandwiches to the m TAs for breakfast, where the TAs can be modeled as $A[1..m]$. Nick plans to use open addressing to ensure that multiple grilled cheeses are not sent to the same TA (so as many TAs as possible get grilled cheeses). Assume that each grilled cheese is equally likely to be sent to any of the m TAs. A probe of grilled cheese i is any time the grilled cheese is sent to a TA (note that due to the nature of open addressing, it is possible for a grilled cheese to make multiple probes before it finds a grilled cheese-less TA). Assume that the probe sequence used to insert for each grilled cheese is equally likely to be any of the $m!$ permutations of the numbers $\{1, 2, ..., m\}$ (uniform hashing assumption). **Justify your answers. Correct answers without justification will receive no credit.**

(a) Suppose you want to send n distinct grilled cheeses, $0 < n < m$, into the group $A[1..m]$ of TAs. What is the expected number of grilled cheeses that require *at least* 2 probes?

(b) Suppose you want to send exactly three distinct grilled cheeses into the group $A[1..m]$ of TAs where $m \geq 3$. In expectation, what is the total number of probes you make during this process?

5. [20 pts] Fridge-ek Patel's Celebrations in Pillow-delphia

In the delightful town of Pillow-delphia, the annual Put a Pillow on Your Fridge Day is the highlight of the year. To celebrate, the charismatic Mayor, Fridge-ek Patel, has declared that every citizen must help adorn the town's fridges with perfectly arranged pillow stacks. Here's the twist: each decorative stack must contain exactly k pillows - no more, no less.

Every citizen comes bearing a positive number of pillows, and the entire town has exactly $k \times x$ pillows, where x is an integer $> n$. Since citizens are a bit shy about decorating alone, they agree to team up in pairs. Each pair will combine their pillows to form as many k-pillow stacks as possible. Note that all k pillows in a stack must be from a single pair of citizens (for example, a stack cannot contain some pillows from one pair and some pillows from another).

Design an algorithm that runs in **expected** $O(n)$ time to determine if it is possible for all the citizens of Pillow-delphia to pair up such that every pair can create some number of perfect k-pillow stacks, yielding exactly x stacks in total.

For example, if each stack must contain 5 items ($k = 5$), and the citizens brought a total of 75 pillows, and the assignment of pillows is as follows:

Citizen:	A	B	C	D	E	F
Pillows:	15	6	5	13	9	27

Here, citizens (A, C) should pair up and create exactly 4 stacks, citizens (B, E) should pair up and present exactly 3 stacks, and citizens (D, F) should pair up and present exactly 8 stacks so your algorithm should return true. If we adjust the assignment such that F brought 28 pillows and E brought 8 pillows, then your algorithm should return false.

6. [25 pts] Go Fly a Kite

For this year's National Go Fly a Kite Day, Aaron is going to do exactly that. Armed with a collection of high-performance kites, each with a unique wind resistance rating, Aaron wants to ensure the perfect kite is chosen for every kind of weather (it varies so quickly in Philly).

Aaron is a proud member of the Aerodynamic Velocity League (AVL), a highly competitive kite-flying organization. To prepare for the aerial fights, Aaron decides to store the kite ratings, naturally, in an AVL tree.

However, as the wind shifts throughout the day, Aaron needs to quickly determine how many kites are viable for certain wind conditions. Extend the AVL tree data structure to implement the following methods for an ordered dictionary D such that each run in $O(\log n)$ time.

CountGEQTo(k_1): compute and return the **number** of items in D with key k such that $k \geq k_1$. Note that k_1 may or may not be in D.

(a) CountGEQTo(k_1): compute and return the **number** of kites in D with wind resistance k such that $k \geq k_1$. Note that k_1 may or may not be in D.

(b) CountInRange(k_1,k_2): compute and return the **number** of kites in D with wind resistance k such that $k_1 \leq k \leq k_2$. Note that k_1 and k_2 may or may not be in D. Assume $k_1 \leq k_2$.

You may use at most $O(1)$ extra storage per node in the tree for each of the methods. If you decide to store any new values/fields, you must briefly describe how those new values/fields are maintained upon insertion and deletion. Show that any modifications you make do not change the AVL Tree's runtimes of INSERT, SEARCH, or DELETE. Note that each item has a unique key and that keys are not necessarily integer values. **Only for part (b), a proof of correctness is not required.**