University of Colombo School of Computing
SCS 1308 - Foundations of Algorithms
Take-Home 08

Instructions

- Try the following questions and upload your answer script as a zip file to the given link in the UGVLE on/before 09th February at 6 pm.
- Note: Rename your zip file with your index number and name. (i.e: indexNo_Name.zip).

Rehashing

1. Write a program to implement rehashing for a hash table. Include functions to:
    - Detect when the load factor exceeds a threshold.
    - Double the size of the table.
    - Reinsert elements into the new table using a new hash function.
2. Given a hash table of size 7, load factor threshold 0.7, and keys [10, 22, 31, 40, 42, 52, 55], write a function to simulate rehashing when inserting keys one by one.

Double Hashing

3. Implement double hashing for collision resolution. Given:
    - Primary hash function: h(k)=k mod 7
    - Secondary hash function: $h_2(k)=5-(k \bmod 5)$Test your implementation with keys [10, 22, 31, 40, 52].
4. Create a program that dynamically handles collisions using double hashing and ensures the hash table does not exceed a given load factor threshold.

Chaining

5. Implement a hash table using chaining with linked lists to resolve collisions. Write functions to:

    - Insert keys.
    - Search for keys.
    - Delete keys.
6. Write a function to create a sorted chain within each bucket of a hash table. Verify its correctness using a set of keys.

Birthday Paradox

7. Write a simulation to calculate the probability of at least two people sharing the same birthday in a group of k people. Verify results for k=23 and k=50.

Perfect Hashing

8. Implement a two-level hash table for perfect hashing. Your program should:
    - Create a primary table using a hash function.
    - For buckets with collisions, create secondary hash tables with unique mappings.
9. Using perfect hashing, hash the set of keys [2, 12, 4, 5, 23, 13, 3] with:
    - Primary hash function: $h(k)=k \bmod 10$
    - Secondary hash function: Design appropriate secondary functions for collisions.