Study Assignment: Hashing and Hash Table Techniques

Objective: This assignment is designed to help you understand the fundamentals of hashing, including different hashing functions, chaining for collision handling, overflow handling strategies, and open addressing methods. By the end of this assignment, you will be able to implement and analyze different hashing techniques and evaluate their efficiency in handling data collisions.

Assignment Sections

1. Hash Functions

- **Objective**: Learn about hashing functions and their characteristics.
- **Tasks**:
  - **Research**: Read about different types of hashing functions such as Division Method, Multiplication Method, and Universal Hashing.
  - **Experiment**: Implement a simple hash function using the division method. Write a small program that hashes integers using your function, then observe how data is distributed across an array of a fixed size.
  - **Report**: Describe how different hashing functions affect the uniformity of data distribution. Include examples and any insights from your experiments.

2. Chaining (Collision Handling)

- **Objective**: Understand chaining as a technique for handling collisions in hash tables.
- **Tasks**:
  - **Implement**: Create a hash table with chaining for collision handling. Use linked lists to store multiple values at the same hash index.
  - **Experiment**: Insert a series of integer keys into your hash table. Identify when and where collisions occur, and observe how chaining handles them.
  - **Analyze**: Calculate the average and maximum chain length after inserting a set of values. Compare the performance of your hash table at different load factors (e.g., 0.5, 0.75, 1.0).

3. Overflow Handling Without Chaining

- **Objective**: Explore alternative techniques for handling overflows without chaining.
- **Tasks**:
  - **Research**: Investigate how overflow handling works without chaining. Look into methods like double hashing and rehashing.
  - **Implement**: Modify your hash table to handle overflows by implementing double hashing. Use a secondary hash function to find new positions when collisions occur.
  - **Compare**: Test and compare the performance of double hashing versus chaining in handling collisions. Record your findings.

4. Open Addressing (Linear and Quadratic Probing)

- **Objective**: Learn open addressing methods for resolving collisions within the hash table.
- **Tasks**:
  - **Linear Probing**: Implement a hash table using linear probing. Handle collisions by moving to the next available slot when a collision occurs.

  - **Quadratic Probing**: Implement a second hash table using quadratic probing. Adjust the probing sequence using quadratic increments.
  - **Experiment**: Insert values and observe how both methods handle collisions. Identify any clustering that occurs with linear probing.
  - **Compare**: Calculate and compare the number of probes required for linear vs. quadratic probing. Record and explain the differences.

Deliverables

- **Code Files**: Submit your implementations for each part (hash functions, chaining, double hashing, linear and quadratic probing).
- **Written Report**: Summarize your findings and insights from each task.
- **Analysis and Comparison**: Include tables or charts showing comparisons in collision handling, probe counts, and efficiency across different load factors for each method.