Instagram Feed Simulation - Project Report

1. Project Overview

This project simulates a simplified version of an Instagram feed. Users can follow others, share posts, view posts in their feed, and like them. The feed prioritizes posts by like count, and secondarily by post ID in lexicographic order. It uses custom-built data structures to manage users, posts, and sorting.

2. System Architecture

The program uses several core classes:

- User: Represents a platform user, including their posts, followings, and interactions.
- Post: Represents content shared by a user.
- FeedManager: Main logic hub, processes commands and manages user feed generation.
- HashTable<K, V>: Custom-built hash table using separate chaining and rehashing.
- MaxHeap: A heap that sorts posts by like count and post ID.

3. Core Functionalities

Key functionalities include:

- User registration and following/unfollowing others.
- Post sharing and like-based feed prioritization.
- Generating personalized feeds with a heap.
- Scrolling through posts and optionally liking them.
- Sorting a user's own posts based on likes.

4. Data Structures

The project does not rely on Java's built-in collections. Instead, it uses:

- HashTable: Implements a dynamically resizing hash table to map user IDs to User objects.
- MaxHeap: Maintains posts in descending order of popularity (likes), using heapify operations for insertion and removal.

5. Sample Execution Flow

1. Users register and follow others.
2. Posts are created and shared.
3. A user generates a feed, seeing top liked posts from followed users.
4. The user scrolls through the feed and likes posts.
5. Posts are re-ranked based on updated like counts.

6. Design Considerations

Custom data structures allow more control and learning about underlying algorithms. Trade-offs include more development effort but better understanding of hash maps and heaps. Complexity is optimized for key operations like insertion, deletion, and sorting.

7. Conclusion

The project showcases how core data structures can simulate real-world app features such as feed ranking and user interactions. Future improvements could include time-based sorting, comment support, and real-time updates.