Module 5: Critical Thinking

Brady Chin

Colorado State University Global

CSC506-1: Design and Analysis of Algorithms

Dr. Dong Nguyen

December 15th, 2024

Module 5: Critical Thinking

In this assignment, we explore the use of hash tables to optimize a social media platform's content recommendation system. Given the vast amount of user data and the need for real-time updates, hash tables offer an efficient way to quickly store and retrieve user preferences and their corresponding recommendations. This implementation simulates the process of generating personalized content recommendations and highlights the benefits of hash tables in enhancing performance and scalability.

Time Complexity

The time complexity of hash tables are O(1). This is the case for adding, updating, removing, or looking up keys since the hash table directly computes the index. The worst case scenario would result in a linear time complexity (O(n)). In this case, the hash table will constantly execute a technique to solve for collisions ().

Justify the Implementation of Hash Tables

Because of the time complexity, it is ideal for the retrieval of content in a social media platform that could have millions of users. Content can be fetched in real time with little to know delays.

Real Life Factors that Impact Performance

Some factors that can impact hash table performance could be the hash function. Poor hash functions could result in collisions meaning multiple keys are mapped to the same index (Fulber-Garcia, V., 2024, March 18).

Data management is another factor that can impact performance. The worst case scenario would be when all data in the hash table have keys that map to the same index. This would result in a linear time complexity (O(n)).

External Factors Influencing the Lower Bound

Even with efficient hash tables and hash functions, there can still be factors effecting the lower bound. Poor network latency can cause delays in fetching content and can effect overall performance. In addition, as the social media platform increases in users and content, there may be a need for rehashing (geeksforgeeks, 2023, March 28).

Corresponding Code

This code is designed to generate personalized content recommendations for each user. Initially, the lack of data and clear direction made it challenging to approach the project. To overcome this, I decided to randomly generate content recommendations for users on the platform.

Hash tables proved to be highly effective for handling large datasets, as their ability to quickly retrieve values using hash keys makes them invaluable in many scenarios.

Conclusion

This project demonstrates the use of hash tables to efficiently manage and generate personalized content recommendations on a social media platform. This assignment highlighted the strength in hash tables when handling large datasets and enabling quick data retrieval.

References

Fulber-Garcia, V., (2024, March 18) *Understanding Hash Tables.* Baeldung. https://www.baeldung.com/cs/hash-tables

geeksforgeeks.org, (2023, March 28) *Load Factor and Rehashing.* https://www.geeksforgeeks.org/load-factor-and-rehashing/