Unit 7: Software Development Life Cycles

e-Portfolio activity

Data Structures Reflection

Task:

Read Dicheva & Hodge (2018). Think about an online system which you use on a daily basis. Consider how it might operate at the back-end using data structures. This will inform our discussion during next week's seminar.

Answer:

An online system that many people use daily is a social media platform, such as Facebook or Twitter. At the back-end, various data structures are employed to manage and optimize the operations of the platform. Here's how some of these data structures might be utilized:

1. **User Profiles and Relationships:**
    - **Hash Tables:** User profiles can be stored in hash tables, allowing for quick access to user data based on unique identifiers (like user IDs or usernames). This enables fast retrieval of user information when someone views a profile or sends a message.
    - **Graphs:** The relationships between users (friends, followers) can be represented as a graph, where each user is a node and each relationship is an edge. This structure allows for efficient traversal and querying of

connections, such as finding mutual friends or suggesting new connections.

2. **Posts and Content Management:**
    - **Linked Lists:** Posts made by users can be stored in linked lists, allowing for efficient insertion and deletion of posts. This is particularly useful for managing a user's feed, where new posts are frequently added and older ones may be removed.
    - **Trees:** A binary search tree (BST) or a more complex data structure like a B-tree can be used to organize posts by timestamps or popularity. This allows for efficient searching, sorting, and retrieval of posts based on various criteria.
3. **Notifications and Messages:**
    - **Queues:** Notifications (like likes, comments, or messages) can be managed using queues. This ensures that notifications are processed in the order they are received, providing a timely and organized way for users to receive updates.
    - **Priority Queues:** For messages, a priority queue can be used to ensure that important messages (like direct messages) are prioritized over less urgent notifications.
4. **Search Functionality:**
    - **Tries:** For implementing search features (like searching for users or hashtags), tries (prefix trees) can be used to efficiently store and retrieve strings. This allows for quick autocomplete suggestions as users type in search queries.

Inverted Indexes: For searching through posts and comments, an inverted index can be utilized. This data structure maps keywords to their locations in the database, enabling fast full-text searches.

5. **Analytics and Insights:**
    - **Arrays and Matrices:** For analytics purposes, data can be stored in arrays or matrices to perform statistical analysis on user engagement, post-performance, and other metrics. This helps in generating insights and improving user experience.

By leveraging these data structures, the back-end of a social media platform can efficiently handle the vast amounts of data generated by users, ensuring a smooth and responsive experience. The choice of data structures directly impacts the performance and scalability of the system, making it crucial for developers to select the appropriate structures based on the specific needs of the application.

References:

Dicheva, D. & Hodge, A. (2018) Active Learning through Game Play in a Data Structures Course. In the Proceedings of the 49th ACM Technical Symposium on Computer Science Education. 834-839.