

# e-Portfolio Activity – Data Structures Reflection

Srihith et al. (2023) emphasise that data structures form the *foundation of efficient computing*, enabling software systems to store, manipulate and retrieve data at scale. As outlined in the article (pp. 155–160), structures such as arrays, linked lists, trees, graphs, hashing mechanisms and B-trees directly determine the speed, memory efficiency and reliability of back-end processes. Reflecting on this, I examined **Nextcloud**, the self-hosted collaboration platform I use daily for synchronising files, managing shared folders and coordinating documents across devices.

Although the interface appears simple, Nextcloud relies on a layered combination of data structures to maintain performance under continuous load. At its core, metadata for files and directories is stored in relational database tables (typically MariaDB), which use **B-trees** for indexing. B-tree indexing is highlighted in Srihith et al. (2023) as a high-performance structure for disk-based retrieval (p. 158), enabling logarithmic-time lookups during operations such as directory listing, file sorting and conflict resolution in synchronisation workflows.

For daily version control, Nextcloud stores revision histories using **append-only logs and linked structures**, avoiding full-object rewrites and reducing I/O overhead. This aligns with the authors' discussion of linked lists as efficient for sequential operations and insertion-heavy workflows (p. 157).

Redis, when enabled, provides a memory cache that relies on **hash tables, sets** and **sorted sets**, all of which are explicitly discussed in the article as examples of constant-time lookup structures essential for high-frequency operations (p. 158). These structures accelerate session handling, file locking and notification queues.

Nextcloud's permission model effectively forms a **graph structure**, where nodes represent users, groups and files, and edges represent access relationships. As Srihith et al. explain, graphs are used extensively for representing real-world relationships and enabling traversal algorithms (p. 156).

This reflection confirms Srihith et al.'s argument that modern systems depend on carefully selected data structures tailored to their operational patterns. Nextcloud's smooth user experience is the result of a sophisticated back-end that leverages these structures for efficiency, scalability and robustness, an insight directly relevant to SDLC decision-making and future seminar discussions.

---

## References:

Srihith, I.D. et al. (2023) *The Backbone of Computing: An Exploration of Data Structures*. *International Journal of Advanced Research in Science, Communication and Technology*, 3(2), pp. 155–163.

Nextcloud Community (2025) *Memory Caching – Administration Manual*. Available at: <https://docs.nextcloud.com>.