

Data Structures Reflection

In software programming, data structures are an essential concept (Dicheva & Hodge, 2018). We can focus on the more fundamental, low-level structures that are used to organize and store data in memory. These include the stack, heap, tree, list, queue, and graphs. Many high-level data structures, such as JSON and XML, use these low-level data structures for parsing and storage.

I want to use Twitter as an example to demonstrate how an extensive system can use these simple and foundational data structures to build their business.

Tree:

Twitter can have a large number of comments, and the system can store them in a tree structure, with the original post as the root and other comments as its children. The system can show comments based on a specific priority because tree data structures are easy to order in a specific sequence. This hierarchical structure allows for efficient storage and retrieval of comments.

Queue:

As an extensive system, Twitter can generate immense log information. If these log data are written to a single location, a queue can help ensure the log output follows their original order. This is very helpful for system analysis when errors occur.

At first glance, many services and businesses in a complex system are built on these simple but efficient data structures. Many algorithms are also implemented based on these data structures. For instance, traversing a tree relies on a stack, while finding the shortest path uses a priority queue.

Brooks, in his famous book "The Mythical Man-Month", mentions, "If you've chosen the right data structures and organized things well, the algorithms will almost always be self-evident. Data structures, not algorithms, are central to programming." (Brooks, 1974). Despite their simplicity, these structures have remained crucial for modern software programming.

References

Brooks, F.P., 1974. The mythical man-month. *Datamation*, 20(12), pp.102.

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.