CS 3303-01 Data Structures

Instructor: Professor Vandelun Amokaha

Name: Ryohei Hayashi

Learning Journal 8

**Overview of the week**

This week, I focused on understanding the terminology related to search and indexing algorithms, as well as studying the characteristics of hash algorithms and self-organizing lists. Specifically, I learned about jump search, dictionary search, and secondary binary search, and recognized the differences between each method. I also studied the use and characteristics of self-organizing lists. Furthermore, I deepened my understanding of Zipf's distribution and considered how it could be applied to improve the efficiency of data searches. Regarding hash algorithms, I implemented open hashing, closed hashing, bucket hashing, linear probing, and collision resolution techniques, learning the advantages and disadvantages of each. I also explored various technical concepts related to index creation, such as entry-ordered files, primary and secondary keys, linear indexes, inverted lists, ISAM, and tree-based indexing methods, including 2-3 trees, B-trees, and B+ trees.

**Personal reflections**

This week’s content was highly technical and required a deeper understanding of algorithms, but it was very rewarding. In particular, while studying and implementing hash algorithms, I made many discoveries about collision resolution methods. I have always been interested in methods to improve database performance, and this week’s content helped deepen my understanding of those techniques. Additionally, learning about Zipf’s distribution was intriguing, and thinking about how it could be applied to actual datasets was very interesting.

**Topics studied in depth**

This week, I delved deeply into hash algorithms and indexing methods. I had the opportunity to understand the differences between open hashing and closed hashing and consider their respective implementations. Additionally, I learned that by using collision resolution techniques like linear probing and bucket hashing, data can be managed more efficiently. Regarding indexing methods, I realized the importance of distinguishing between entry-ordered files, primary keys, and secondary keys. With tree-based indexing, I gained a deeper understanding of how structures like B-trees and B+ trees improve database search speeds.

**Reflection on the entire course**

Looking back at this entire course, I feel that my understanding of the fundamental concepts of data structures and algorithms has significantly deepened. From the start of the course, the importance of how data structures and algorithms form the foundation of efficient problem-solving was consistently emphasized. Although the course focused heavily on theoretical aspects, I also learned how these concepts could be applied to real-world problem-solving through various case studies and assignments.

One of the most memorable aspects of the course was using visual tools to understand the operations of data structures and algorithms through animations. This helped me intuitively grasp how data structures like stacks, queues, linked lists, hash tables, trees, and graphs work. I also learned the importance of time complexity and space complexity when analyzing algorithms and gained a foundational understanding of designing efficient algorithms.

What was particularly helpful during the course was the gradual progression from recursion and basic data structures (stacks, queues, lists) to binary trees, non-binary trees, internal and external sorting, and finally, search and indexing algorithms. Each week's content built upon the previous one, giving me the feeling that my understanding of more complex data structures and algorithms was gradually solidifying.

I was especially impressed by the topics of binary and non-binary trees. Understanding binary trees helped me recognize the importance of data structures that significantly influence search efficiency and algorithm performance. Learning about non-binary tree structures also provided me with the skills to apply algorithms to more complex datasets. Finally, studying sorting and indexing algorithms allowed me to think about how they could be applied to databases and file systems.

Throughout this course, I felt that the gap between theory and practice was effectively bridged. I not only learned about algorithms but also developed the ability to think about how to design efficient programs. The sections on file processing and external sorting, in particular, stood out as crucial skills for real-world system design.

Moving forward, I plan to apply what I have learned in more practical projects and utilize my knowledge of data structures and algorithms in my work. Additionally, continuing to learn about algorithm optimization and new techniques to handle complex datasets will be my focus moving forward.

Word Count: 682

References

1. Shaffer, C. (2011). *A Practical Introduction to Data Structures and Algorithm Analysis.* Blacksburg: Virginia. Tech.