A Lumberjack’s Guide to Red-Black, B+ and Prefix Trees

Jheyne Cordeiro, Alessandro Loi, Nate Seon, Meiru Zhang

Team 07, CS 622 Discrete Math|Algorithm Computing

Master of Sciences in Computer Science

City University of Seattle

[deoliveirapantacord@cityuniversity.edu](mailto:deoliveirapantacord@cityuniversity.edu), [loialessandro@cityuniversity.edu](mailto:loialessandro@cityuniversity.edu), [seonnate@cityuniversity.edu](mailto:seonnate@cityuniversity.edu), [zhangmeiru@cityuniversity.edu](mailto:zhangmeiru@cityuniversity.edu)

**Abstract**

This academic work aims to explore three essential tree data structures in computer science: Red-Black Trees, B+ Trees and Prefix Trees. In computing, data structure is fundamental for the good performance of algorithms and applications. A Red-Black Tree is a type of binary search tree with self-balancing, this is because after insertions and deletions, it guarantees O (log n) operating times. It is also known for quickly storing and retrieving requested information and ensuring that operations are completed within a specified time frame. The B+ Tree is a data structure based on balanced search for trees, such as The Red-Black Tree. Due to poor performance using large amounts of memory, other binary trees may be limited, but B+ Tree has a much better structure as it can handle large amounts of data. A Prefix Tree, also known as a Try, is a data structure used in computer science to efficiently store a set of strings. They are useful for search operations, such as finding words in a dictionary or performing autocomplete. We used motivating examples, and we were careful to verify these trees with comparison and data structures covered in the course highlighting their advantages and compensations for each one. Through a practical use case, we had the opportunity to demonstrate the algorithmic efficiency of these structures. This study aims to provide valuable information on selecting the appropriate data structure for specific computational tasks.

**Keywords:** Red-Black Trees, B+ Trees, Prefix Trees, Data Structures, Tree Algorithms, Algorithmic Efficiency.

**1.** **INTRODUCTION**

Computer science is dynamic, and choosing the right data structure is very important, it would be like selecting the best strategy for a sports team. This article aims to highlight three essential tree data structures, such as Red-Black Trees, B+ Trees and Prefix Trees. Thus, as in sports where good strategies are used, we can say that the Red Black Trees resemble a well-coordinated basketball team with balance, maintaining the team's formation under pressure. As for B+ trees, we can compare them to the manual of a football team that is designed with strategic movements, this is because B+ trees have like the precision and quick responses that a tennis player presents during a play. By comparing these structures through sporting strategies, we will demonstrate how they really work, but in a more engaging way. This study provides practical and simple examples of what it would be like in the real world, contributing to a better understanding of more effective data structures in the computational world.

**9.** **REFERENCES**

Aibin, M. (2024). Tries Prefix Trees.

https://[www.baeldung.com/cs/tries-prefix-trees](http://www.baeldung.com/cs/tries-prefix-trees)

Emertex. (n.d.). *Data Structured Project | Red-Black Tree.* Red Blacj Tree (RBD).

[https://www.emertxe.com/embedded-systems/data-structures/ds-projects/red-black-tree /](https://www.emertxe.com/embedded-systems/data-structures/ds-projects/red-black-tree%20/)

GeeksforGeeks. (2023). Introduction of B-Tree.

<https://www.geeksforgeeks.org/introduction-of-b-tree-2/>