ARENA OF RATINGS: MATCHMAKING ENGINE

Real-Time Matchmaking Engine with Persistent Data & Visual CLI

Muhammad Hammad Shakeel
BS Artificial Intelligence | IMSciences

1. THE ENGINE: INTRODUCTION

The **Arena of Ratings (Ultimate)** is a sophisticated matchmaking engine designed for competitive gaming. Unlike standard lists, it uses a **Binary Search Tree (BST)** with subtree sizing to ensure **O(h)** efficiency.

New in Ultimate Edition:

- **Persistent Storage:** Save/Load player data to files.
- **Visual Interface:** Color-coded CLI with tree topology.
- **Smart Metrics:** Real-time Duel Distance calculation.

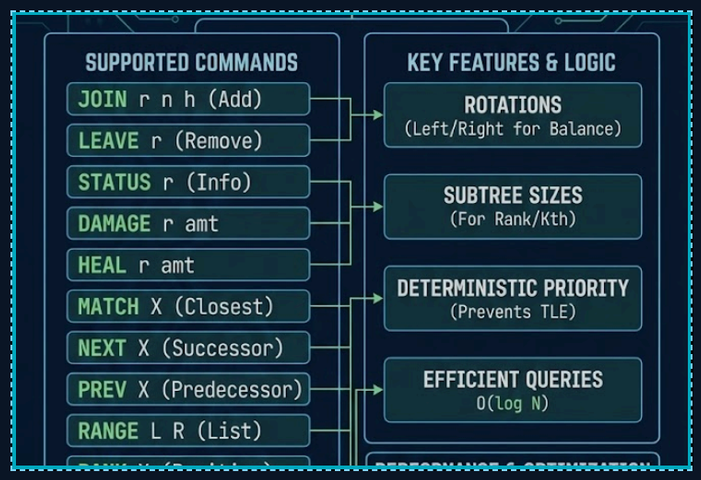
2. CORE DATA STRUCTURE

The system is built on a custom Node structure handling both game data and structural balancing metrics:

```
struct Node {
    int rating;      // The Key (Ordered)
    string name;     // Player ID
    long long hp;    // Dynamic Health
    int sz;          // Subtree Size
(Ranking)
    Node *left, *right;
};
```

Rating: Determines tree position.
Subtree Size (sz): Enables finding the "K-th" best player in milliseconds.

3. MAIN VISUAL: DETERMINISTIC TREAP



"A glowing cybernetic Binary Search Tree on a dark blue background."

4. ALGORITHMIC LOGIC

Recursive Efficiency: All major operations use recursion to traverse the tree efficiently.

- _insert(): Adds players while updating subtree sizes.
- _duel(): Calculates distance using Lowest Common Ancestor (LCA).
- _printDirectoryStyle(): Custom recursive function drawing the tree sideways in the terminal for debugging.

5. SUPPORTED COMMANDS

The engine supports a robust Command Line Interface (CLI):

JOIN / LEAVE	Dynamic player mgmt
MATCH <X>	Finds nearest rated opponent
DUEL <A> <B>	Calculates graph distance
SAVE / LOAD	Persist state to .txt
VISUAL	Draw Tree topology
STATS	Diagnostics & Metrics

6. PERFORMANCE METRICS

- **Time Complexity:** O(h) for Search, Insert, Delete.
- **Space Complexity:** O(N) space efficiency.
- **Real-Time Feedback:** Uses ANSI color codes (\033[32m) for immediate visual feedback (Green for Success, Red for Errors).

7. CONCLUSION

The **Match Making Engine** transforms a standard BST into a production-ready engine. By integrating file persistence and visual diagnostics, it bridges the gap between theoretical data structures and real-world software application.