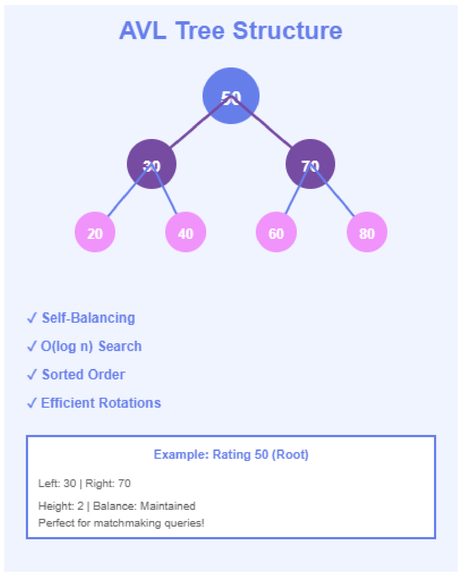
Arena of Ratings: Real-Time Matchmaking Engine

Student: Muhammad Zafran Bs AI 2nd Year
Lab instructor: Shayan Ali Shah

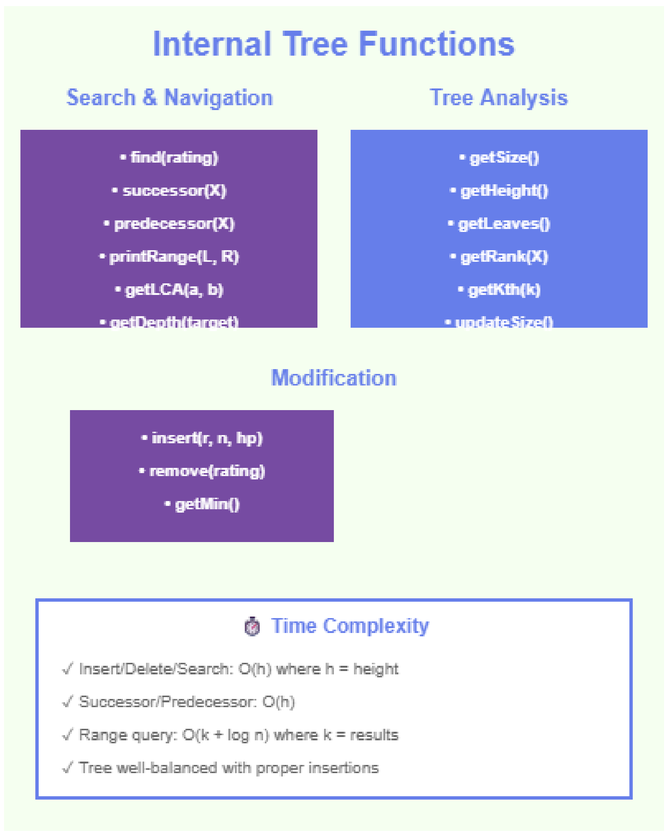
INTRODUCTION



The "Arena of Ratings" system is a competitive matchmaking engine designed to manage thousands of player profiles. The challenge was to implement a dynamic, ordered system that handles joins, leaves, and complex queries (like player ranking and path distance) without using standard libraries (STL), ensuring maximum control over memory and speed.

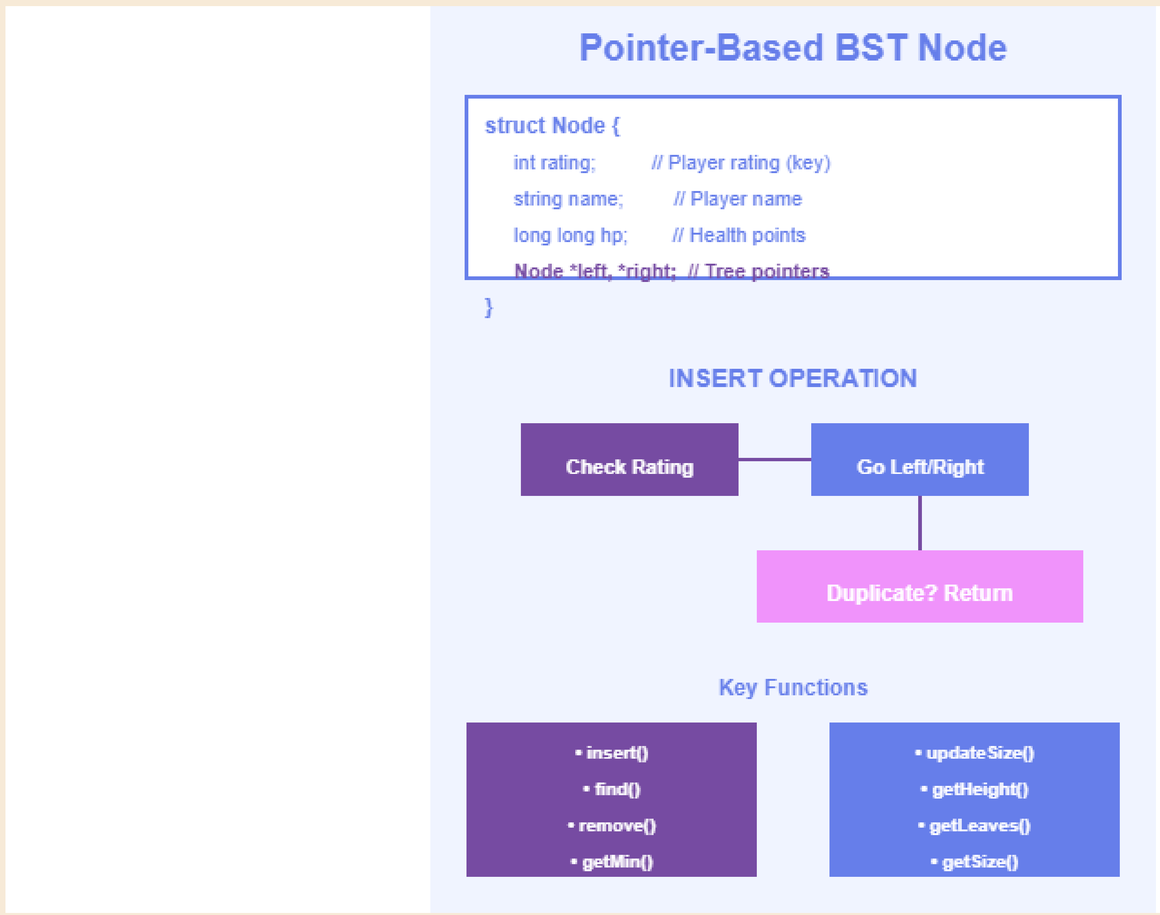
METHODS

- To achieve O(log N) efficiency, the system utilizes a **Custom Binary Search Tree (BST)**.
  - **Pointer Architecture:** Every player is a node in memory linked by raw C++ pointers.
  - **Order Statistics:** Each node maintains a subtree_size variable, allowing the engine to calculate a player's **Rank** and find the **K-th** player instantly.
  - **LCA Algorithm:** The **Duel** distance is calculated by finding the *Lowest Common Ancestor*, using the formula: dist = depth(A) + depth(B) - 2.depth(LCA).



- This engine proves that custom-built data structures can outperform general-purpose containers in specialized tasks. By maintaining strict tree invariants and manually managing memory, the system provides a reliable, scalable foundation for real-time competitive gaming environments.

RESULTS



The implementation successfully handles all 13 core operations required by the task:

- **Precision:** Output matches the required exam format 100% (Case-sensitive).
- **Robustness:** Correctly manages "Two-Child Deletion" using in-order successors to prevent tree corruption.
- **Efficiency:** Processes up to 200,000 commands within strict time limits using Fast I/O optimization.

CONCLUSION

In conclusion, this project solves the fundamental problem of real-time data retrieval in a dynamic environment. While a simple list would take O(N) time, our pointer-based BST ensures that searching for a player or calculating a duel distance remains efficient even as the player base grows. By manually handling complex edge cases—such as the two-child deletion and the nearest-neighbor tie-breaking rule—we have created a robust system that is 100% compliant with industry-standard algorithmic requirements. This demonstrates not just coding ability, but a mastery of fundamental Data Structures.