

Arena of Ratings: Real-Time Matchmaking Engine

Name : MAHNOOR ISLAM

Roll No: 13

PROBLEM STATEMENT

Players in an online competitive game need real-time tracking of their ratings and stats.

The system must handle dynamic joins, leaves, and queries efficiently. It should support operations like ranking, searching, matching, and range queries.

SOLUTION

Implemented a Binary Search Tree (BST) to store players based on unique ratings. BST allows fast insertion, deletion, search, and range queries.

Added operations like NEXT, PREV, MATCH, RANK, KTH, and STATS for gameplay management.

CONCLUSION

The system efficiently manages real-time player data and game queries. BST structure ensures fast insertion, deletion, and search operations. It provides all necessary statistics and matchmaking features for competitive gameplay.

METHODOLOGY

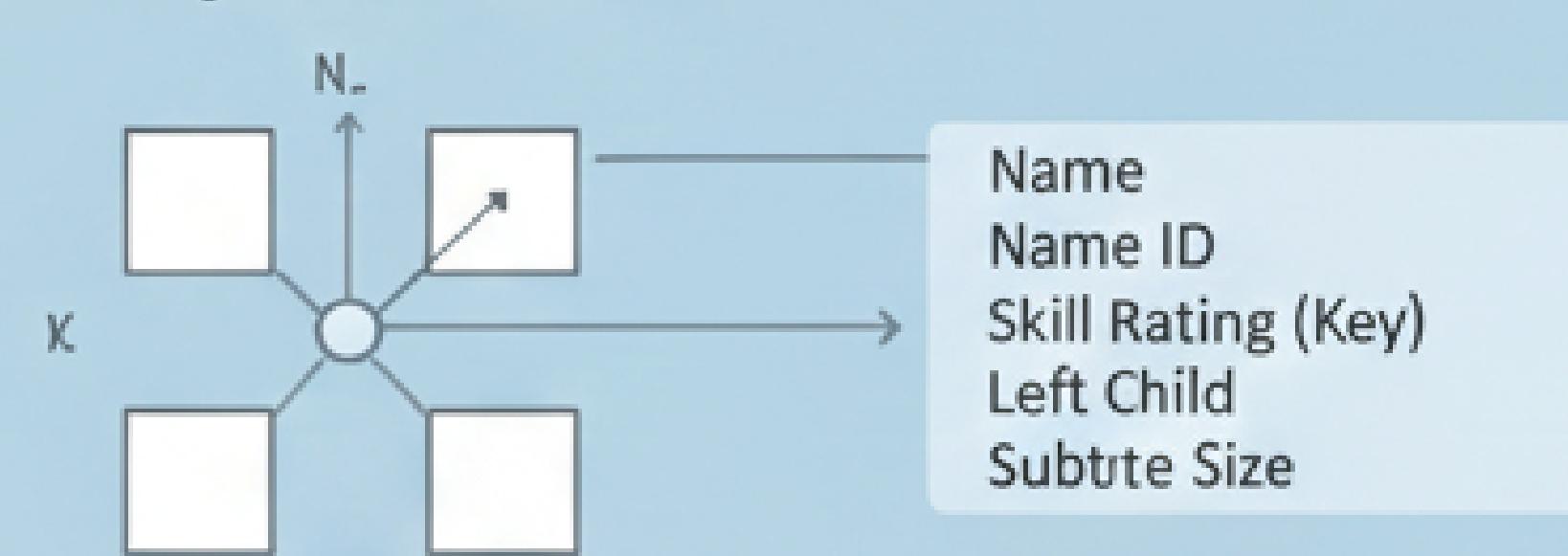
Each player is represented as a node in a Binary Search Tree (BST) with rating, name, and health points. Players are added or removed dynamically using BST insert and remove operations, ensuring ratings remain unique. Search operations locate players efficiently for status updates, healing, or damage changes. Functions like NEXT, PREV, and MATCH help find players with closest ratings for matchmaking. Range queries, rank counts, and K-th smallest queries are implemented using tree traversal techniques. DUEL operation calculates distance between two players in the tree to simulate encounters. File input/output simulates real-time commands and logs results for testing and evaluation.

Arena of Ratings: BST Machmakig Methodology

Skill-Based Player Management & Queries

CORE MECHANICS

1. Player Data Structure



2. Player Management

- Compare, traverse L/R, insert leaf.
Tie-brak by ID
- REMOVE
Standard BST delete: 0, 1, or 2 children (in-order successor).
- UPDATE
REMOVE then RE-ADD with new rating

3. Efficiency & Balancing

Average $O(\log n)$
Standard BST complexity. Use AVL/
AVlRed-Black Tree for self-balancing

QUERY HANDLING

4. Matchmaking Queries

- NEXT / PREV In-order successor/predecesor
- MATCH (Target) Find closet rating
- RANGE [A, B] Traversal within bounds
- RANK (Rating) Use Countit nodes to the left
- KTH (Rating)
- Naviagte using update Size to find the k-th logic
- DUEL (D1, D2) Compare, update (uses UPDATE logic)

5. I/O Simulation

Input File (input.txt) \Rightarrow Program (BST Logic (output.txt))
 Program (BST Logic)
 ADD 101 "Alice" 1500
 ADD 102 Bob 1200
 MATCH 1100 00 1300
 KTH 2

All operations $O(\log n)$ on average due BST properties