

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.

