**Project Title: Red-Black Tree Implementation and Performance Analysis**

**Project Description:** In this project, students will implement a Red-Black Tree data structure from scratch in a programming language of their choice (preferably a language that supports pointers or references for ease of implementation). **You can use the posted PDF to get some basic background on red-black trees.** The Red-Black Tree implementation should include insertion, deletion, and searching methods.

**Tasks:**

1. **Implementation of Red-Black Tree:**
   * Students will write code to implement the Red-Black Tree data structure, ensuring that it adheres to the properties of a Red-Black Tree, such as color properties, balancing, and ordering.
   * They should implement methods for insertion, deletion, searching, and traversal.
2. **Test Suite:**
   * Develop a comprehensive test suite to verify the correctness of the Red-Black Tree implementation. This suite should include both unit tests for individual methods and integration tests to ensure the overall functionality of the tree.
3. **Performance Analysis:**
   * Conduct a performance analysis of the Red-Black Tree implementation. This analysis should include measuring the time complexity of key operations (insertion, deletion, searching) for various input sizes.
   * Compare the performance of the Red-Black Tree with other data structures like AVL trees or regular binary search trees. Students should analyze how the Red-Black Tree performs under different scenarios and input distributions.
4. **Visualization (Optional):**
   * Optionally, students can create a visualization tool to display the Red-Black Tree structure graphically. This can help better understand the tree's balancing and operations.
5. **Documentation and Presentation:**
   * Document the implementation details, including explanations of algorithms, optimizations, and the rationale behind design choices.

**Deliverables:**

* Source code of the Red-Black Tree implementation.
* Test suite demonstrating the correctness of the implementation.
* Performance analysis report, including time complexity measurements and comparisons with other data structures.
* Documentation covering implementation details and optimizations.

**Benefits:**

* This project allows students to understand better balanced binary search trees, particularly Red-Black Trees, and their implementation challenges.
* Students will enhance their programming skills by implementing complex data structures and algorithms.
* Through performance analysis, students will learn to evaluate and compare different data structures' efficiency, which is crucial for making informed design decisions in software development.