

Deliverable 2

Muhammad Ali Rehman

RED NO. 2023094, 2023372

COUSRE CODE: ES-221

COURSE NAME: DATA STRUCTURE AND ALGORITHM

1. Introduction

Git Lite is a simplified version of GitHub that demonstrates core version control and data organization functionalities using C++ and Data Structures & Algorithms (DSA) concepts. This project utilizes an AVL Tree for efficient management of commits, users, or repositories.

2. Why AVL Tree?

An AVL tree is a self-balancing Binary Search Tree where the height difference (balance factor) between left and right subtrees is at most 1 for all nodes. This guarantees $O(\log n)$ time complexity for all basic operations.

Time Complexity of AVL Tree Operations:- Insertion: $O(\log n)$ - Deletion: $O(\log n)$ - Search: $O(\log n)$ - In-order Traversal: $O(n)$

3. Core Features of Git Lite

The following features are supported in Git Lite:- User account creation and lookup- Repository creation and indexing- Commit addition and history management- Search commits/users efficiently using AVL trees

4. Key Functions Required

Essential functions in C++:- insertCommit(Commit c)- deleteCommit(string commitID)- searchCommit(string commitID)

- displayHistory()- insertUser(User u)- searchUser(string username)

5. Suggested C++ Classes

class Commit:- string commitID- string message- string timestamp

class User:- string username- string email- AVLTree<Commit> commitHistory

6. Additional C++ Concepts Used

Git Lite also incorporates various C++ standard and system libraries for file and data handling:

- <fstream>: For file input/output operations
- <sstream>: For parsing and formatting text
- <iomanip>: For controlling input/output formatting
- <windows.h>: For system-specific operations (e.g., clearing screen)
- <filesystem>: For directory and file management task

CODE FILE FOR DELIVERABLE 2 WILL BE ATTACHED