Reflection on Assignment 1

Working on this module was a valuable learning experience involving data structures and algorithms. My assigned problem was to determine whether a binary tree contains duplicate values, and if so, to identify the one closest to the root. Initially, I felt challenged, particularly in deciding the correct traversal strategy to efficiently locate duplicates. I found myself deciding between Depth-First Search (DFS) and Breadth-First Search (BFS). I ultimately chose a breadth-first approach, which allowed me to ensure I found the closest duplicate efficiently.

As best practices tell me to do when coding, the process led me to consider adding capability of processing edge cases, such as trees with only one node or those with multiple levels but no duplicates. It was rewarding to see my understanding deepen as I tried different solutions and optimized the code.

Reflection on Partner's Assignment Review

Reviewing my partner's assignment was a constructive experience. It gave me a chance to practice code review, which is a key aspect of collaborative software development. I appreciated the opportunity to understand someone else's approach to solving a similar problem, and it made me realize the variety of methods one can employ to solve algorithmic challenges. My partner used a set-based approach to find missing numbers, which was effective. Due to the potential memory overhead when dealing with large datasets, there was an improvement opportunity for scalability. This prompted me to think of alternatives such as boolean arrays, which could achieve the same outcome with reduced memory usage. This review process solidified my confidence in understanding the problem but also exercised a skill we have not used yet; the ability to communicate constructive feedback clearly to another professional.