Assignment#2: In-memory DBMS using Trees

In this assignment, the "database" will be a simple, memory-based (non-persistent, no filing required) structure that allows storing and managing data. There will be tables that store records.

Each "table" should store records with three fields:

- ID (integer, unique identifier)
- Name (string)
- Age (integer)

The database tables will be represented using the below different tree structures:

1. Binary Search Tree (BST)
2. AVL Tree
3. B-Tree

Each tree structure will independently store the same table data, serving as an "index" for organizing and managing the records. By implementing the table with the given type of trees, you can examine how each structure affects the performance of various database operations.

For example:

For a Binary Search Tree (BST) for a particular table, the records in that table (E.g. BST_TABLE) will be organized according to the BST rules, with each record represented by a node in the tree.

Similarly, if you use an AVL Tree for the table (E.g. AVL_TABLE), the records in that table will follow AVL Tree properties, ensuring that the tree remains balanced after each insertion or deletion.

Each tree type should support common database operations.

- Insert: Adding a new record to the table.
- Search: Finding a record by ID.
- Update: Changing the details of an existing record.
- Delete: Removing a record by ID.

Performance Analysis

Implement timing functionality to measure the performance of each operation (Insert, Search, Update, Delete) for each tree structure. For each operation, measure the average time for a dataset of various sizes (e.g., 1,000, 10,000, and 50,000 records). For example, for a fixed size of records, say 10,000 what is the average search time of 20 search queries in all the three types of tables?

Note: (You will need to create a separate function to generate dummy data of the given size).

Present results in a table or graph and write a short analysis of the time complexity observed versus expected for each data structure.