Nama: Valentino Marcell

NIM: 2602084104

The Question is

1. What are the difference between binary tree and b-tree?
2. What are the difference between AVL Tree and Red Black Tree? In what case would you want to use a red black tree over an AVL tree and vice versa?

The answer

1. A binary tree is a tree-like structure with at most two children per node, while a B-tree is a self-balancing tree with multiple children per node. Binary trees lack balance and are used for simple hierarchical structures, while B-trees maintain balance and are optimized for disk access, commonly employed in file systems and databases. B-trees minimize disk access through a balanced structure and a larger number of keys per node, while binary trees are suited for in-memory operations.
2. AVL Trees and Red-Black Trees are both self-balancing binary search trees, but they differ in their balancing strategies. AVL Trees maintain strict balance factors for each node, resulting in faster lookup times but slower modifications due to frequent rotations. Red-Black Trees use a coloring scheme to ensure balance, allowing for more efficient insertions and deletions at the cost of slightly slower lookups. The choice between AVL Trees and Red-Black Trees depends on the specific requirements of the application, with AVL Trees being preferred for read-heavy scenarios and Red-Black Trees for dynamic or write-intensive applications.

In what case would you want to use a red black tree over an AVL tree and vice versa?

1. When the program have more frequent insertion and deletion operations than search operations
2. When the program have memory constraints
3. When the program prioritize faster search operations