Name: Bryan Orville Audric

NIM: 2602160570

No 4a

The difference between Binary Tree and B Tree lies in the number of keys and number of children that a node can have. In Binary Tree data structures, a node can only hold a single key and have two children node (left & right). On the other hand, B Tree data structures allow some nodes to hold more than one keys and can have more than two children node.

No 4b

Both AVL Tree and Red Black Tree are similar in a way that both are a balanced tree. The main difference of both trees would be the number of rotations that each tree has. AVL Tree has higher number of rotations compared to Red Black Tree.

Speaking about properties each data structures has different set of rules:

Each node in AVL Tree should have a balanced factor of > 1 or < -1 (The sign ‘- ‘is there to track whether the tree is leaning towards right or left side). Balance factor is essentially a difference of height between both children node. The data structure tracks each height of the node when insertions happen. It will either rotate to the right or left whenever a violation occurs. Violation occurs when a balanced factor of a node reached 2 or -2 (lower or higher).

Each node in RBT is tagged as black or red. A violation occurs whenever a node is consecutively red or there is unequal amount of black node from each path from root node to leaf node. It will either rotate to the right or left whenever a violation occurs and colour flip will occur afterwards. It could also do a colour flip on the node that currently cause the violation. Both fixes that were mentioned earlier will occur when a certain condition is reached. For instance, Colour flip occurs when the parent sibling of a node is red. Rotation will occur when the parent sibling of a node is black. Rotation will occur after the rotation is done.

In general, it is preferable to use a Red Black Tree data structure when there is a lot of insertion and deletion needs to happen. It is due to fewer rotation that the Red Black Tree has compared to AVL Tree, hence it allows faster insertion and deletion features. On the other hand, AVL Tree is preferred when searching often happens. It has slower insertion and deletion process compared to Red Black Tree because it thoroughly balanced every node. Due to this, the tree are more balanced compared to Red Black Tree, hence it offers faster searching process.