Task-4

**Step 1) In your own words, describe what a Binary Search Tree (BST) is.**

Binary Search Tree is a data structure that represents a hierarchical structure. Binary Trees have a root node on top and children nodes. Every node can be a subtree. The difference between Binary Tree and Binary Search Tree is that the value of the left node in BST is always less than the value of its parent’s node, and the value of the right node in BST is always bigger than the value of its parent’s node. Because of that, it is very easy to search, insert, and delete values in the data structure.

**In addition, describe two important properties of a BST: depth and height. How are they different?**

In BST, depth is the distance, the number of edges from the root node to the particular node. Height, in BST, is the height of the tree. It refers to the longest path from the root node to a leaf node(a node without any children).

**Step 2) In your own words, describe how an algorithm to find an item in a Binary Search Tree works.**

When searching in a Binary Search Tree, the algorithm starts looking from the root node. Then, if the value we are looking for is less than the node we are on, we go to its left node and so on. But if it’s bigger than the node, we go to its right node and continue. We repeat this process until we find the item or until we reach a leaf node where the item is not found.

**Step 3) In your own words, describe what a balanced BST is.**

A balanced BST is a BST where the heights of the left and right subtrees of any node differ by at most one. The tree is almost symmetrically balanced around its root node.

**Step 8) With the newly balanced BST, how many steps does it take at most to find an existing item in the search tree?**

The root node of the newly balanced BST is the middle element of the sorted list of Binary Search Tree object. The height of the left and right subtrees of any node in this newly balanced BST differs at most one level. It is just like performing a binary search on a sorted list.

The time complexity of searching would be O(log n).

The reason is that at each level of the balanced BST, the number of nodes usually doubles. So it goes, 1 node, 2 nodes, 4 nodes, and 2^x nodes where x represents the level. So, the maximum number of steps required for the search is logarithmic.