Assignment 7

Satyanand
14EC10049

RED BLACK TREE

A red-black tree is a binary search tree in which

1. each node has a color (red or black) associated with it (in addition to its key and left and right children)
2. the following 3 properties hold:
3. (root property) The root of the red-black tree is black
4. (red property) The children of a red node are black.
5. (black property) For each node with at least one null child, the number of black nodes on the path from the root to the null child is the same.

Operations on a Red-Black Tree

As with the binary search tree, we will want to be able to perform the following operations on red-black trees:

1. insert a key value (insert)
2. determine whether a key value is in the tree (lookup)
3. remove key value from the tree (delete)
4. print all of the key values in sorted order (print)

Red-black tree operations are a modified version of BST operations, with the modifications aiming to preserve the properties of red-black trees while keeping the operations complexity a function of tree height.

Red-black tree insertion:

Inserting a node in a red-black tree is a two step process:

1. A BST insertion, which takes O(log n) as shown before.
2. Fixing any violations to red-black tree properties that may occur after applying step 1. This step is O(log n) also, as we start by fixing the newly inserted node, continuing up along the path to the root node and fixing nodes along that path. Fixing a node is done in constant time and involves re-coloring some nodes and doing rotations.

Accordingly the total running time of the insertion process is O(log n). Figure 7 shows the red-black tree in figure 5 before and after insertion of a node with value 4. You can see how the swap operations modified the tree structure to keep it balanced.

Red-black tree deletion:

The same concept behind red-black tree insertions applies here. Removing a node from a red-black tree makes use of the BST deletion procedure and then restores the red-black tree properties in O(log n). The total running time for the deletion process takes O(log n) time, then, which meets the complexity requirements for the primitive operations.

Red-black tree retrieval:

Retrieving a node from a red-black tree doesnt require more than the use of the BST procedure, which takes O(log n) time.