Cs301 -a3

Oğuzhan Özdemir (Student)

Problem 1

The RBTs running time is O(h), which is h = lgn, to find black height of the RBT we need to compute from leaf to x. Because of the propriety number 5 that every path from x to a leaf will have the same number of black nodes. To compute the black-height of a given node we only need one sides of black height. To compute black height does not increase the complexity which is lgn. Insertion to tree does not effected by this because we count from leaf to top so adding just effects the ancestor of this node.

Problem 2

To compute the depth of a given node in an RBT insertion or deletion case cannot be constant. Depth is height of the node but in many cases the tree will re balance it self. For example if we insert a key which only balance itself to become root the all depth will be changed n times. In problem 1 we can maintain the complexity becouse the changes only effects nodes ancestors but this case it can effect every node.