CS301 Assignment 3 Answers

Uğur Öztunç 28176

Problem 1

The black height of a node is the number of black nodes from that node to a leaf. And, if we consider the property of Red-Black Trees, which says that the number of black nodes on the path from any node to all its descendant leaves must be equal, we can say that looking at just one of the two subtrees of a node will be enough to compute its black height. In addition to these, due to the fact that the insertion operation effects the nodes through the root (ancestor nodes) and as mentioned above, black height of a node is calculated by looking from leaf to that node, the time complexity of insertion operation, which is O(logn), will not be affected.

Problem 2

The depth of a node is the distance of that node from the root. In order to maintain the additional field of depth of nodes, we need to look from the given node through the root, not from the leaves to the given node as in the previous example. This results that since the insertion affects the ancestor nodes, it will not take constant time to maintain the attribute of depth, since all nodes of the RBT may be affected in case of an insertion. To give an example for this situation, if we insert a new node to a RBT and, in order to preserve the properties of RBT, there were such rotations performed that newly added node became the new root. In this case all nodes in RBT are affected as well as the depth fields of them. Consequently, augmenting the depths of nodes as additional attributes in the nodes of a RBT will increase the time complexity of insertion operation, since maintaining the depth fields of all nodes will take non-constant time.