Homework № 3

Lal Verda Çakır, Istanbul Technical University 11/01/2021

Complexity [10 pts] Write down the asymptotic upper bound for the insertion and search operations of RedBlack Tree for worst case and average case with detailed explanations..

Red black tree is almost a balanced complete tree so the level count is always smaller than $log_2 n + 1$ therefore the insertion complexity of worst case is $O(logn)$. In the worst case we will be inserting at the end so we need to traverse till the deepest point. In the insertion other than going till the end coloring, restructring, changing colors and handling double red operations are happening. Handling double red is $O(logn)$ in worst case and the other operations are $O(1)$ so in conclusion worst case insertion is $O(log_2 n)$

In the insertion we actually search the element we are going to add our new node to so the search algorithm is $O(logn)$ as well.

Because our depth is not worst than $log_n n + 1$ average case is the worst case so the complexities are the same($O(logn)$)

RBT vs BST [5 pts] Compare Red-Black Tree with Standard Binary Search Tree in your own words.

Both of them are binary trees that if the node has a left child that childs data is smaller than the nodes data and if that node has a right child that childs data is greater of equal to the nodes. The difference between these to is mainly because rbt is a self balancing tree so it heiht is bounded by $log_2 n + 1$ so that search algorithm takes a logarithmic time but in the bst for example in the worst case the inserted data may be already sorted so we can end up with long chain of nodes at the right so in worst case bst's height is bounded by n so the search complexities are $O(logn)$ to $O(n)$