

Report for assignment 7

Manne Naga Nithin (14EE10026)

March 3, 2016

Inserting

1. Function Outline

We perform a normal binary search tree insertion firstly, then we balance the colours by checking the corresponding colours and rotating and colour shifting, wherever necessary.

We make the node red initially, then we check the black heights and red red rule and rotate or colour shift accordingly.

case 1: If both the children are red and one of their children is red.

The solution is to make the child black and the root red and send the violation up the tree.

case 2: If one child is black and the other is red and the closest child of this is also red.

The way to solve this is to convert it into case three either by a right or left rotation.

case 3: If one child is black and the other is red and the farthest child of this is also red.

Make a single right or left rotation as required to bring down the colour violation.

2. Time Complexity

$$T(n) = O(\log(n))$$

Deleting

1. Process

First, we flush out all the processes from the tree into a linked list, we operate on the first node and reinsert all the elements back into a new tree.

If the process has completed execution, we do not need to reinsert it back into the tree, we just insert the remaining elements.

2. Time Complexity

$$T(n) = O(n)$$