Chandan A.M
CS025

18/11/20

DATE | | PAGE:

7) Red Black Tree Insertion :

```
void RBTree :: insert (int &data) {
     Node *pt = new Node (data);
     root = BSTInsert (root, pt); // BST insertion
      fix_violation (root, pt);
}


void RBTree :: fix_violation(Node *&root, Node *&pt) {
    Node *parent_pt = NULL;
    Node * grand_parent_pt = NULL;
    while((pt!=root) && (pt -> color != BLACK) &&
                pt -> parent -> color == RED)) {
         parent_pt = pt -> parent;
         grand-parent_pt = pt -> parent -> parent;
```

```
if (parent_pt == grand_parent_pt -> left) {
    Node * uncle_pt = grand_parent_pt -> right;
if( parent_pt = grand_parent_pt -> left) {
        Node* uncle_pt != NULL &&
                uncle_pt -> color == RED ) {
        grand_parent_pt -> color = RED;
        parent_pt -> color = BLACK;
        uncle_pt -> color = BLACK;
        pt = grand_parent_pt;
}
else{ if( pt == parent_pt -> right ) {
        rotateleft( root, parent_pt) ;
        pt = parent_pt;
        parent_pt = pt -> parent;
    }
    rotateRight ( root, grand_parent_pt);
    swap (parent_pt -> color , grand_parent_pt
                                    -> color));
    pt = parent_pt;
}
}
```

```
else {
    Node * uncle_pt = grand_parent_pt -> left;
    if ( (uncle_pt != NULL) &&
         (uncle_pt -> color == RED)) {
        grand_parent_pt -> color = RED;
        parent_pt -> color = BLACK;
        uncle_pt -> color = BLACK;
        pt = grand_parent_pt;
    }
    else {
        if (pt == parent_pt -> left) {
            rotate right (root, parent_pt)
            pt = parent_pt;
            parent_pt = pt -> parent;
        }
        rotate left (root, grandparent_pt,
        swap (parent_pt -> color,
                grand_parent_pt -> color);
        pt = parent_pt;
    }
  }
}
root -> color = BLACK;
}
```