18/11/2020 Red Black Tree Insertion

Sahana L
1BM18CS089

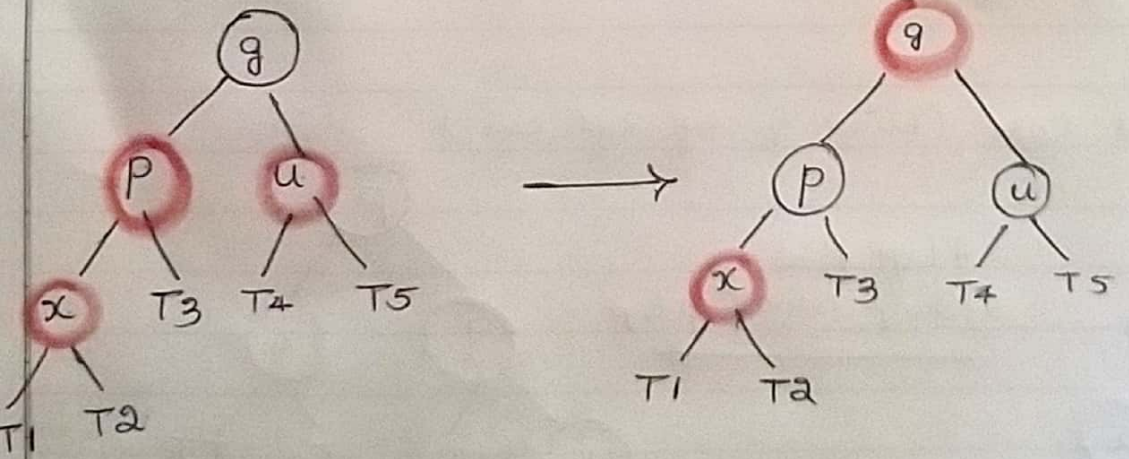
```
enum Color {RED, BLACK};
struct Node{
    int data;
    bool color;
    Node *left, *right, *parent;

    Node(int data)
    {
        this->data = data;
        left = right = parent
```

Algorithm

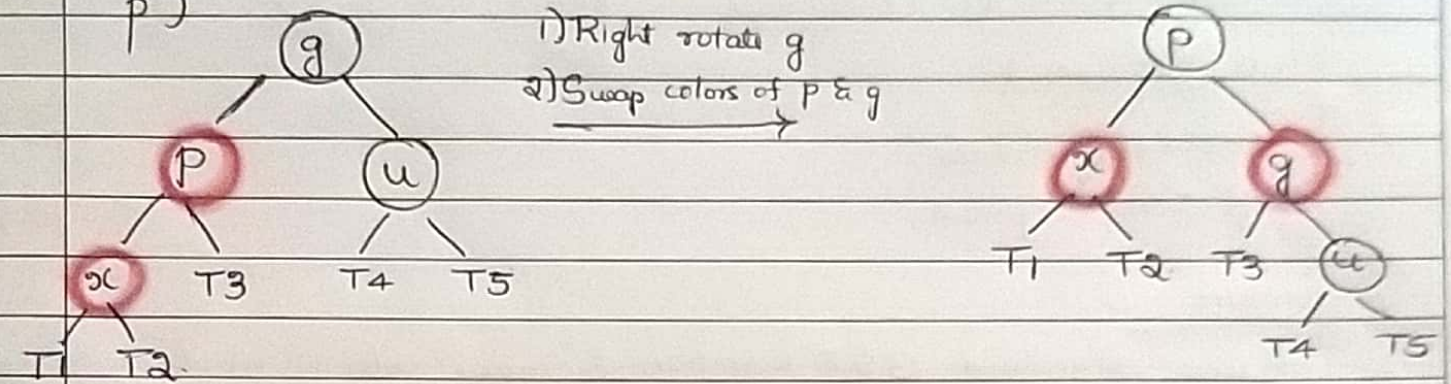
1. Perform standard BST insertion & make color of newly inserted node as RED.
2. If x is root, change color of x as BLACK or x is not root.
3. If color of x's parent is not BLACK or x is not root, then
    1. If x's uncle is RED
        (i) Change color of parent & uncle as BLACK
        (ii) color of grandparent as ~~red~~ RED
        (iii) Change x = x's grandparent & repeat steps 2 & 3 for new x

Uncle red

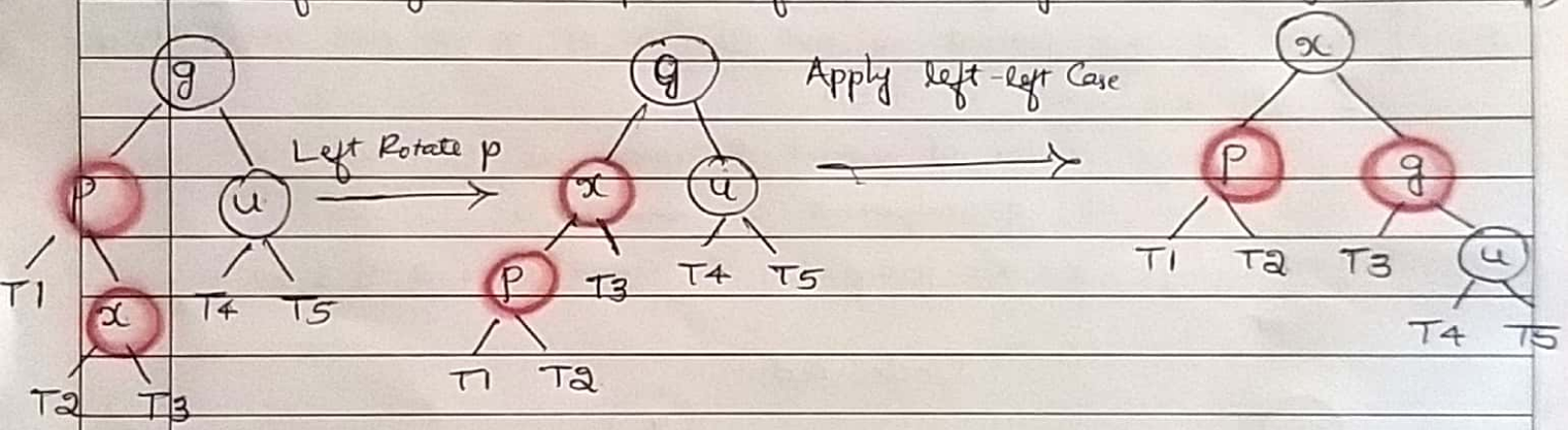


2. If x's uncle is BLACK, then there can be 4 configurations for x, x's parent (p) and x's grandparent (g).

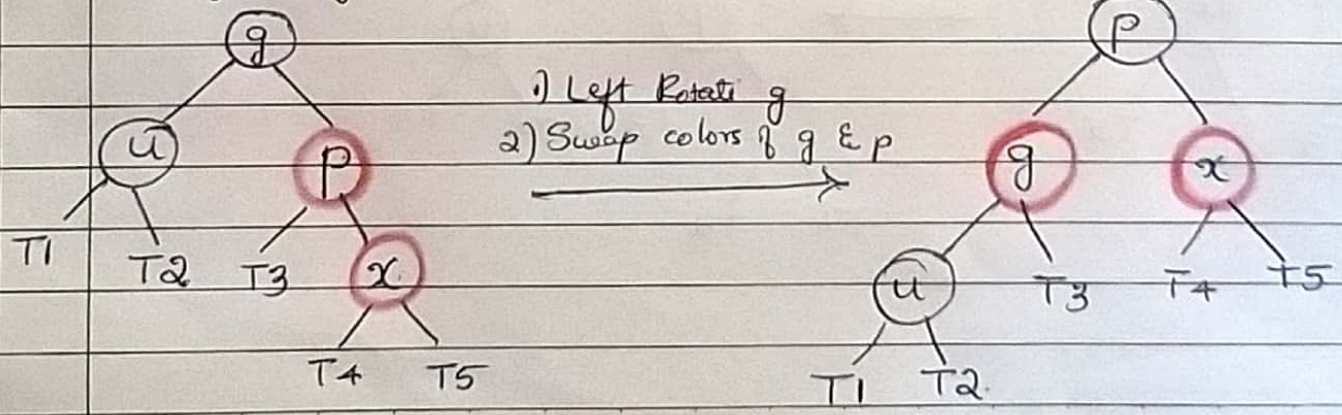
(i) Left - Left Case (p is left child of g and x is left child of p)



(ii) Left - Right Case (p is left child of g and x is right child of p)



(iii) Right Right Case (Mirror of left left case)



(iv) Right - Left Case