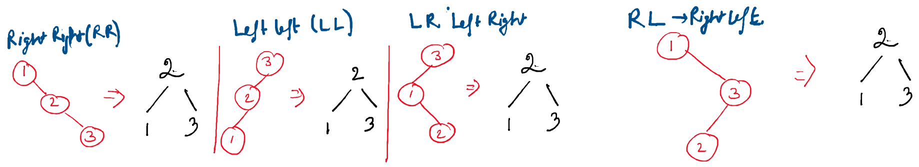
RBT U section

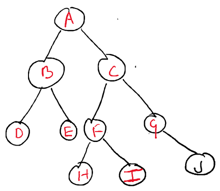
06 February 2025 22:56

Rotation In BST



Left child = Minimum
Right child = Max.
Root = Remaining one.

Terminology – Grandparent, Parent, Uncle.



Node	Parent	Grand Parent	Uncle.
J	G	C	F
I	F	C	G
H	F	C	G
D	B	A	C
E			
F			

RBT Property – A Red-Black Tree is a type of binary search tree providing efficient insertion, deletion, and lookup operations.

Properties of Red-Black Trees

1. **Node Color**: Each node is either red or **black**.
2. **Root Property**: The root of the tree is always **black**.
3. **Red Property**: Red nodes cannot have red children (no two consecutive red nodes on any path).
4. **Black Property**: Every path from a node to its descendant null nodes (leaves) has the same number of **black** nodes.
5. **Leaf Property**: All leaves (NIL nodes) are **black**.

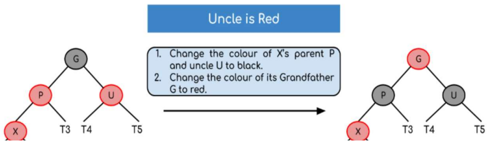
Step 1:

- Insert the new node with color **red**.

Step 2:

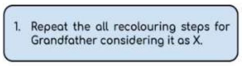
- Case 1: Node is the Root
  - Recolor it to **black**.
- Case 2: Red-Red Violation (Parent and newly inserted node are both red)
  - Case 2.1: Uncle is Red
    - Recolor the **parent** and the **uncle** to **black**.
    - Recolor the **grandparent** to **red**.
    - Repeat the fix-up process from the **grandparent**.
  - Case 2.2: Uncle is Black or Null
    - Perform **rotations** to balance the tree.
    - Recolor the nodes accordingly.

G = Grand Parent , P = Parent
U = Uncle , X = New Node to Insert.



Uncle is Red

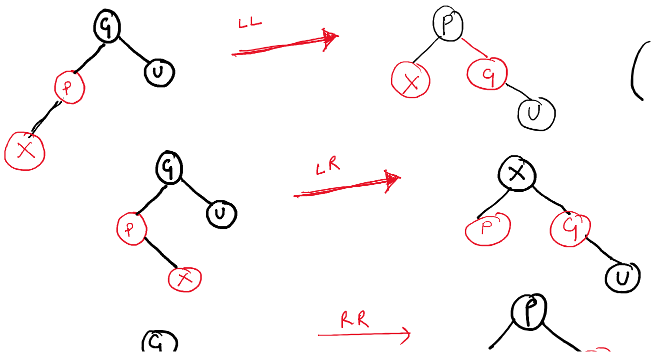
1. Change the colour of X's parent P and uncle U to black.
2. Change the colour of its Grandfather G to red.



1. Repeat the all recolouring steps for Grandfather considering it as X.

→ Change P, G, U

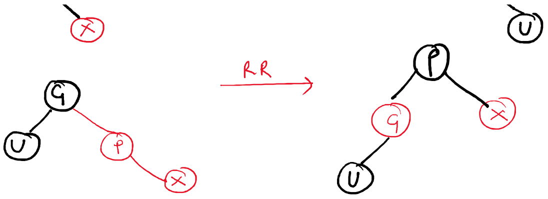
Uncle is Black / NULL



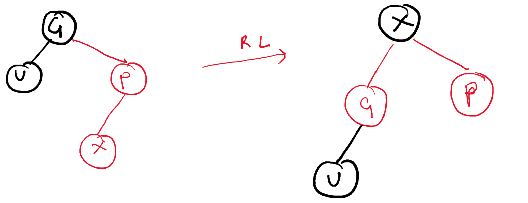
(P, G swap color)

(X, G swap color)

(P, G swap color)

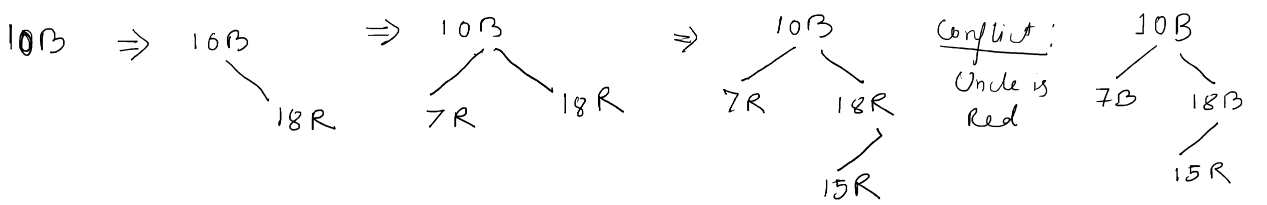


(P, G swap color)



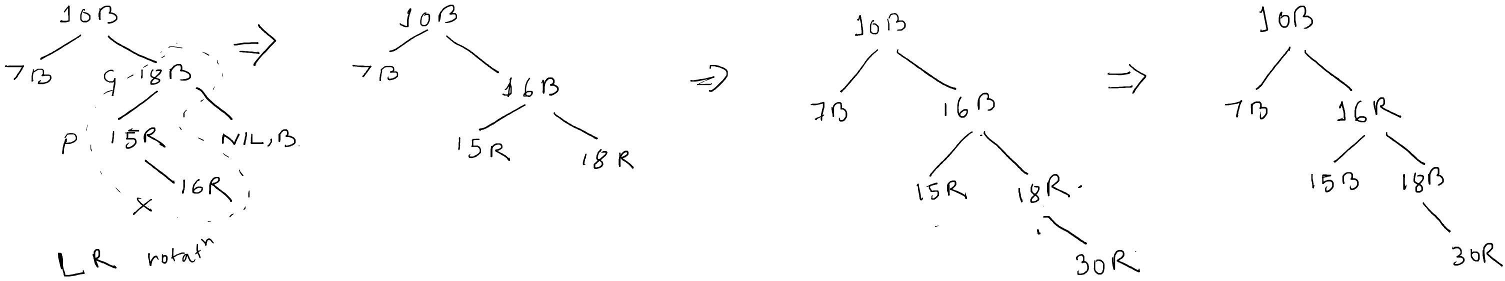
(x, G swap color)

10 18 7 15 16 30 70



Conflict: Uncle is Red

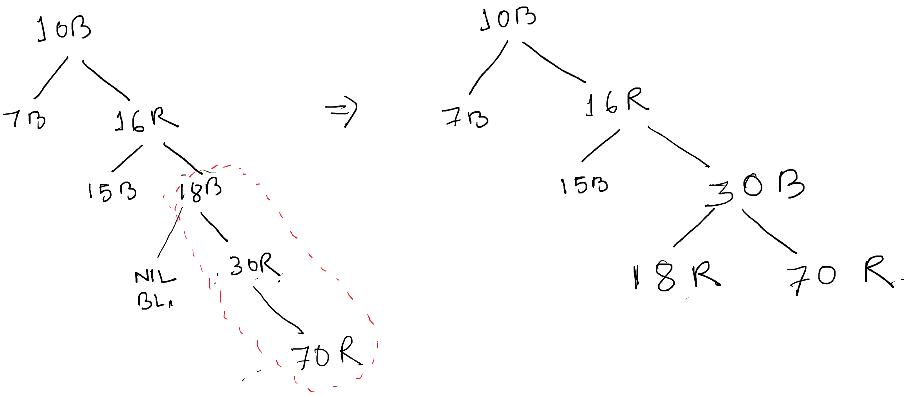
But Root
10RX



LR rotation

x, G swap

Uncle = Red.



Uncle = Black / NIL

P, G swap color