

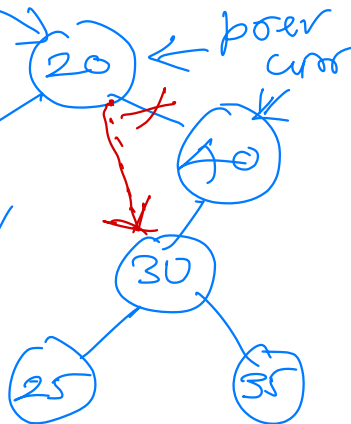
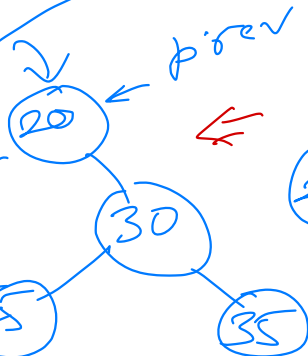
Fix to delete a elem from BST.

4.1 // curr node has only one child

if (curr is root)
Make curr's child as root
STOP

4.1 Replace prev's child from curr to curr's child.

4.2 STOP



AVL Tree Rotation/Balancing

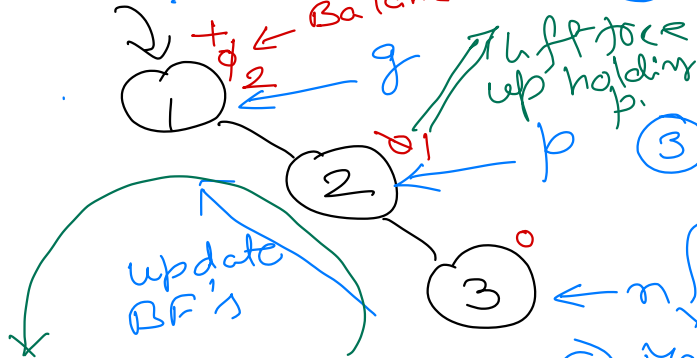
$$\text{Balance Factor} = |h_L - h_R|$$

$$\leq 1$$

Insert:

1 2 3 4 5 6

not null



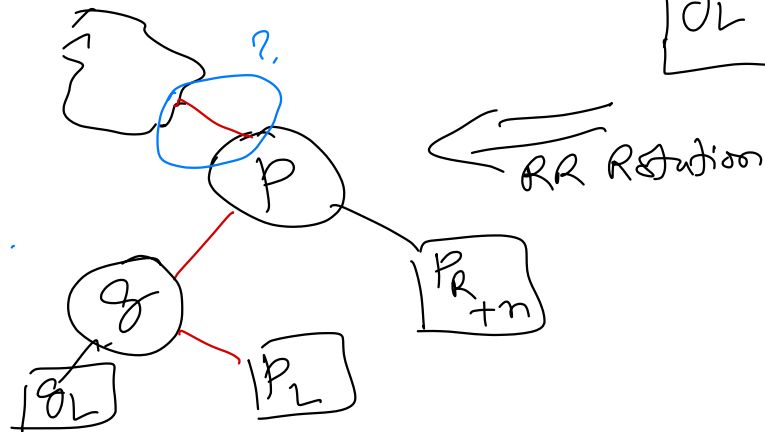
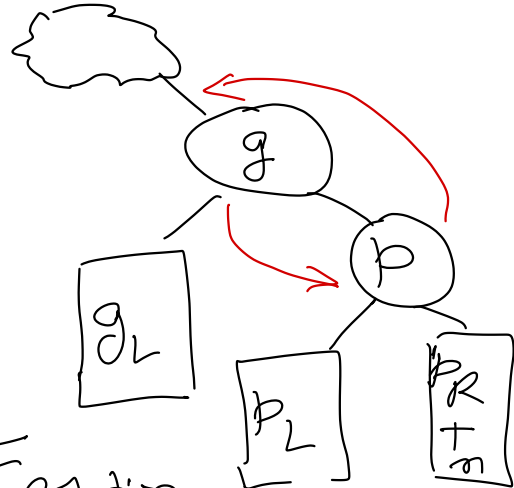
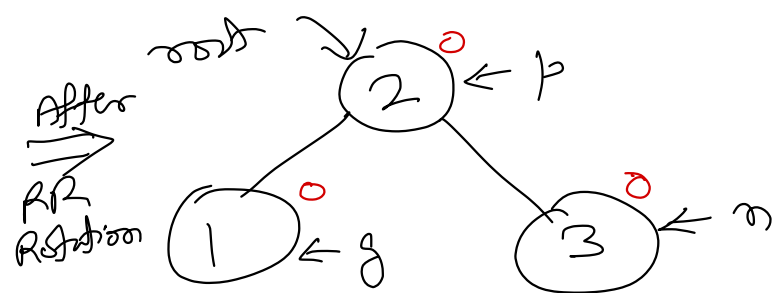
① n = new node added to BST.

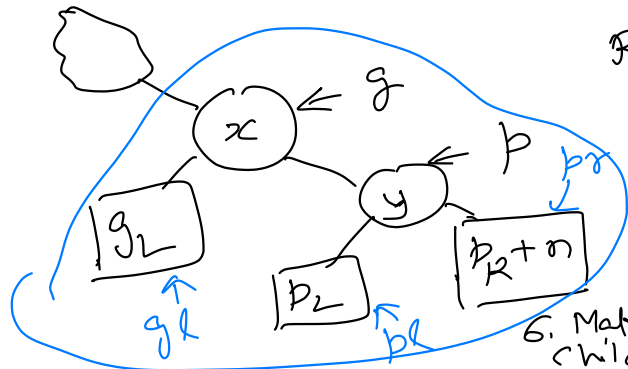
② g = nearest parent of new node, n , having incorrect Balance Factor

③ Take 2 steps from g towards n .

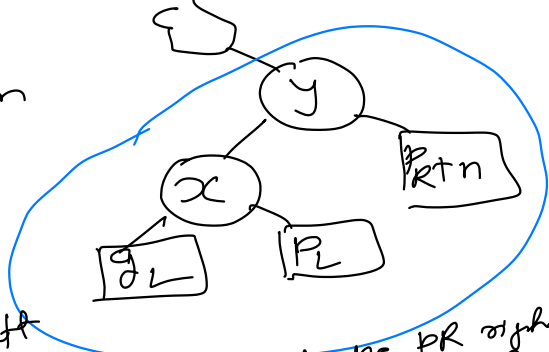
RR Rotation
LL LR RL ← other Rotations

④ Mark p = right child of g .
⑤ Left rotate tree around p .



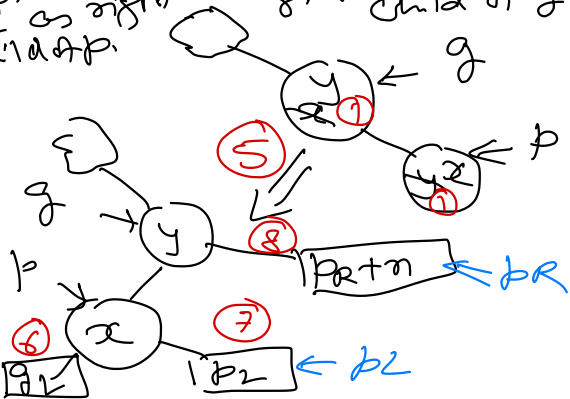


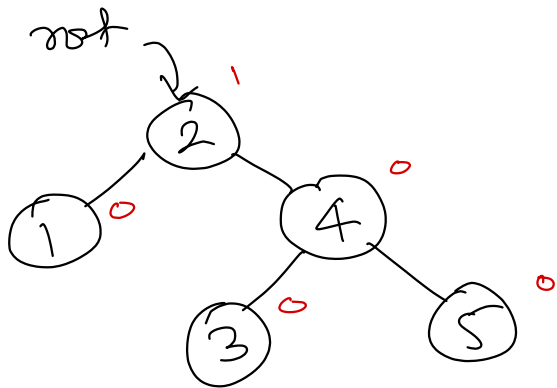
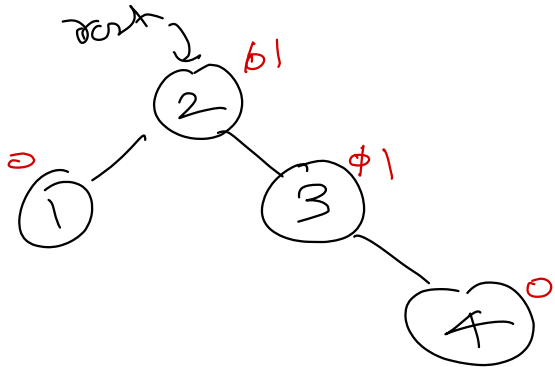
RR
Rotation



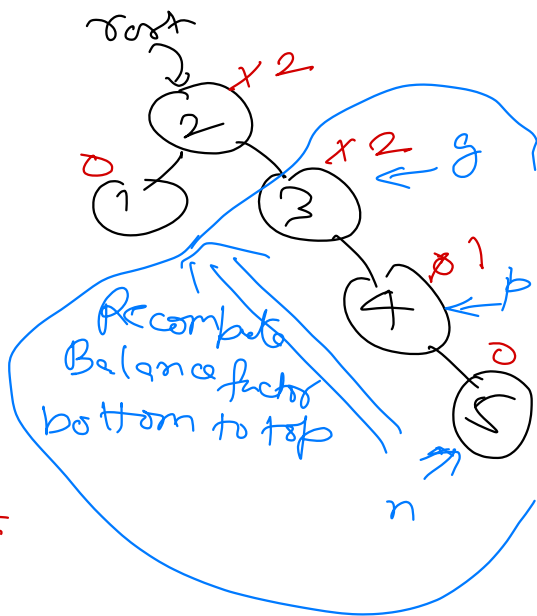
RR(g, p)

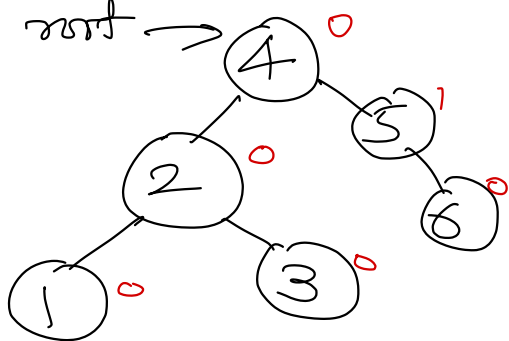
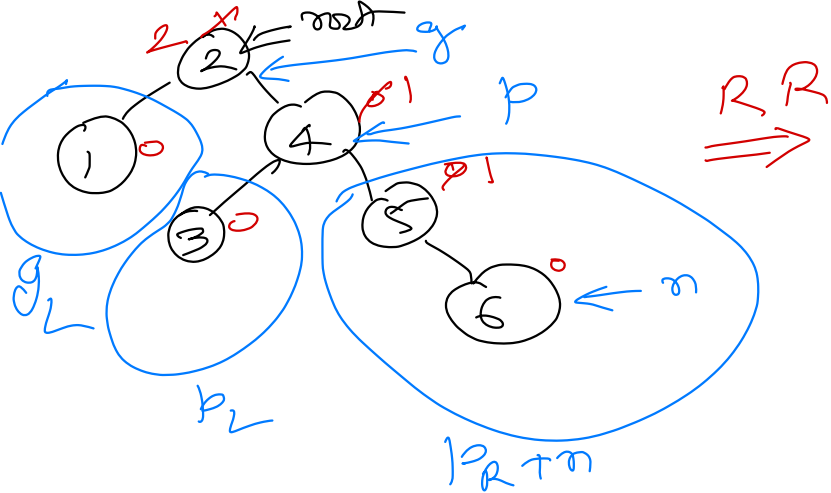
1. Swap data of node g & p.
2. Set g_L to g's left child.
3. Set p_L to p's left child.
4. Set p_R to p's right child.
5. Make p as left child of g.
6. Make g_L left child of p.
7. Make p_L as right child of p.
8. Make p_R right child of g.



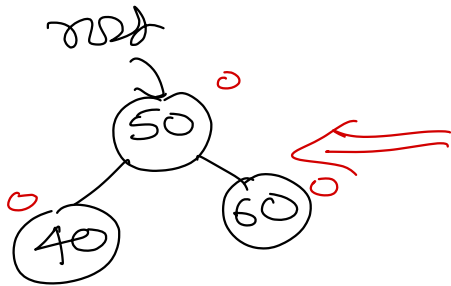
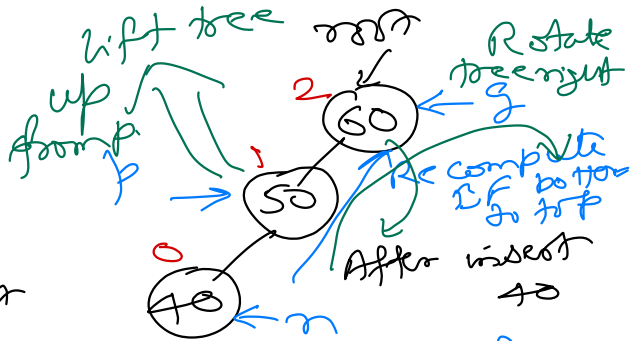
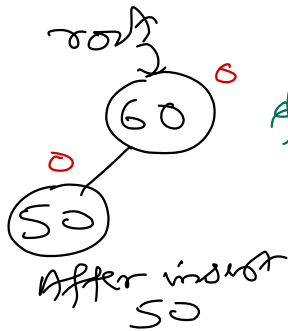
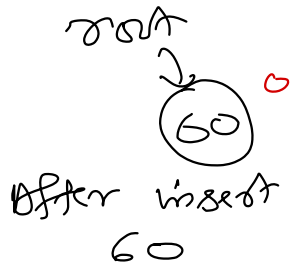


RR



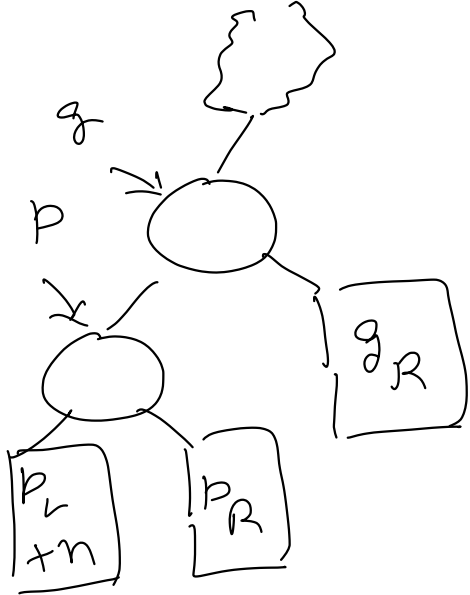


Insert 60 50 40 30 20 10

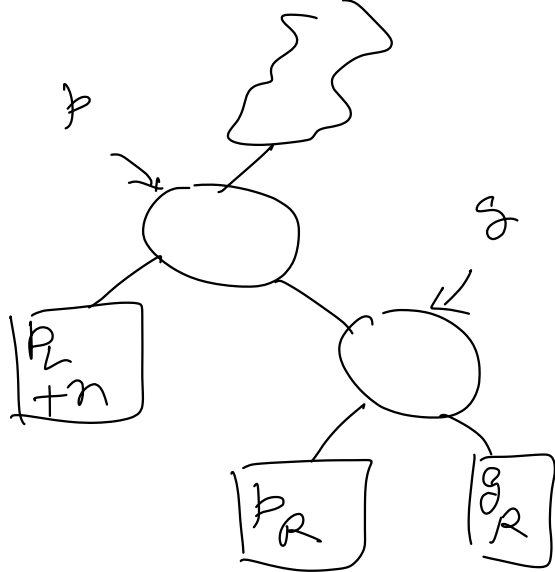


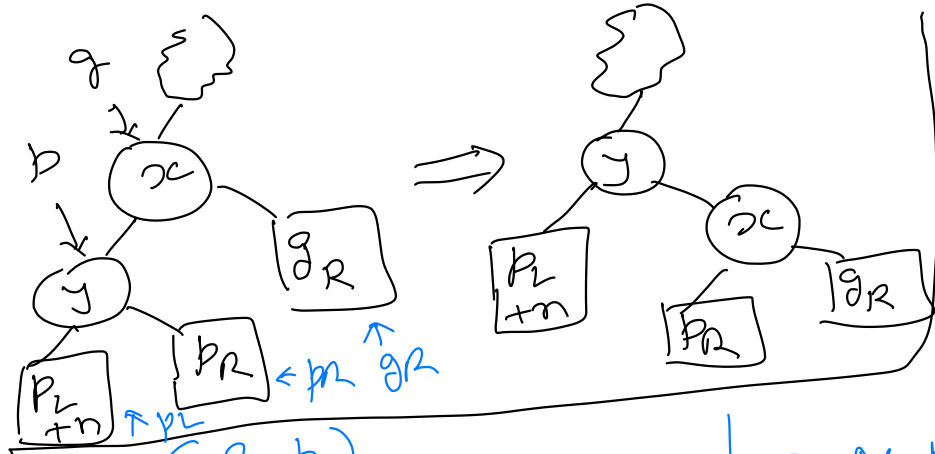
Take 2 steps from g towards n

For LL Rotation
 Mark p a left child of g
 Perform LL Rotation around p.



\Rightarrow
LL
Rotation



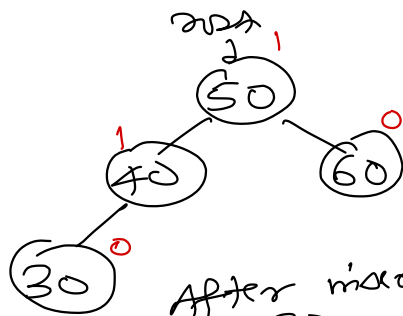
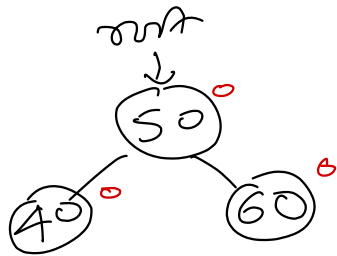


LL (g, p)

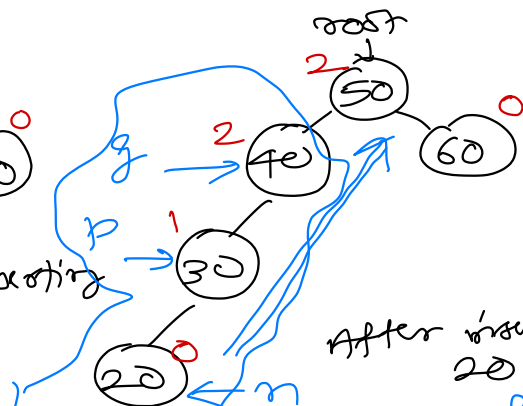
1. Swap data of g & p.
2. Set pL to left child of p.
3. Set pR to right child of p.
4. Set gR to right child of g.



5. Make p as right child of g.
6. Make pL as left child of g.
7. Make pR as left child of p.
8. Make gR as right child of p.



After inserting 30



After inserting 20

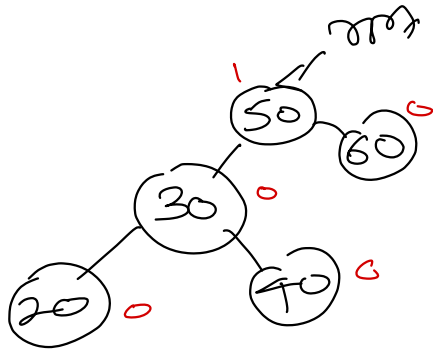
Take 2 steps from g towards n.

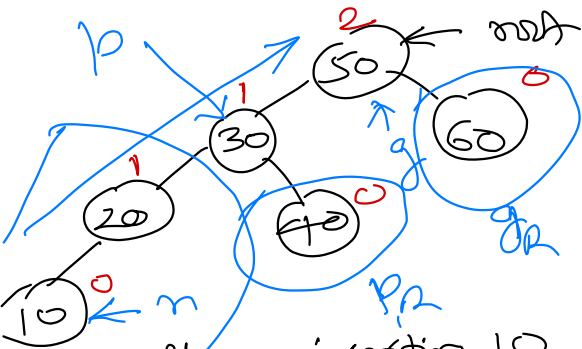
LL Rotation

Mark left child of g as p.

Perform LL rotation around p.

LL rotation





After inserting 10

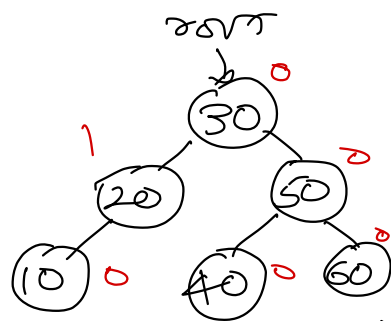
p_2

Take 2 steps
from g
towards n

Rotation

Mark left
child of g as p.

Do LL rotation
around p.

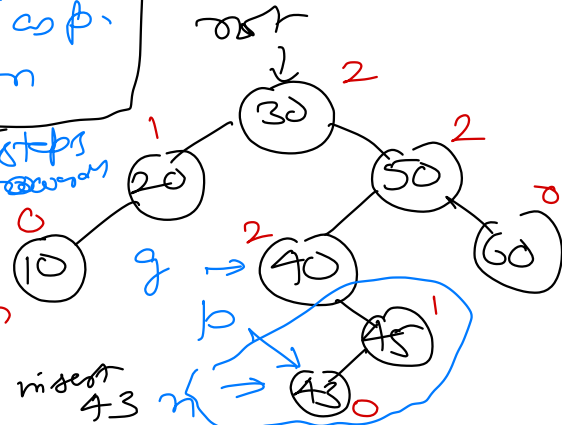


After insert 45

Take 2 steps
from g towards
n

RL
Rotation

After insert 43



In RL Rotation. to mark p

→ Take two steps from g.

Step 1. Go to right child of g.

Step 2. Go left of right child of g.

RL Rotation around p

⇒ first LL rotation around p.

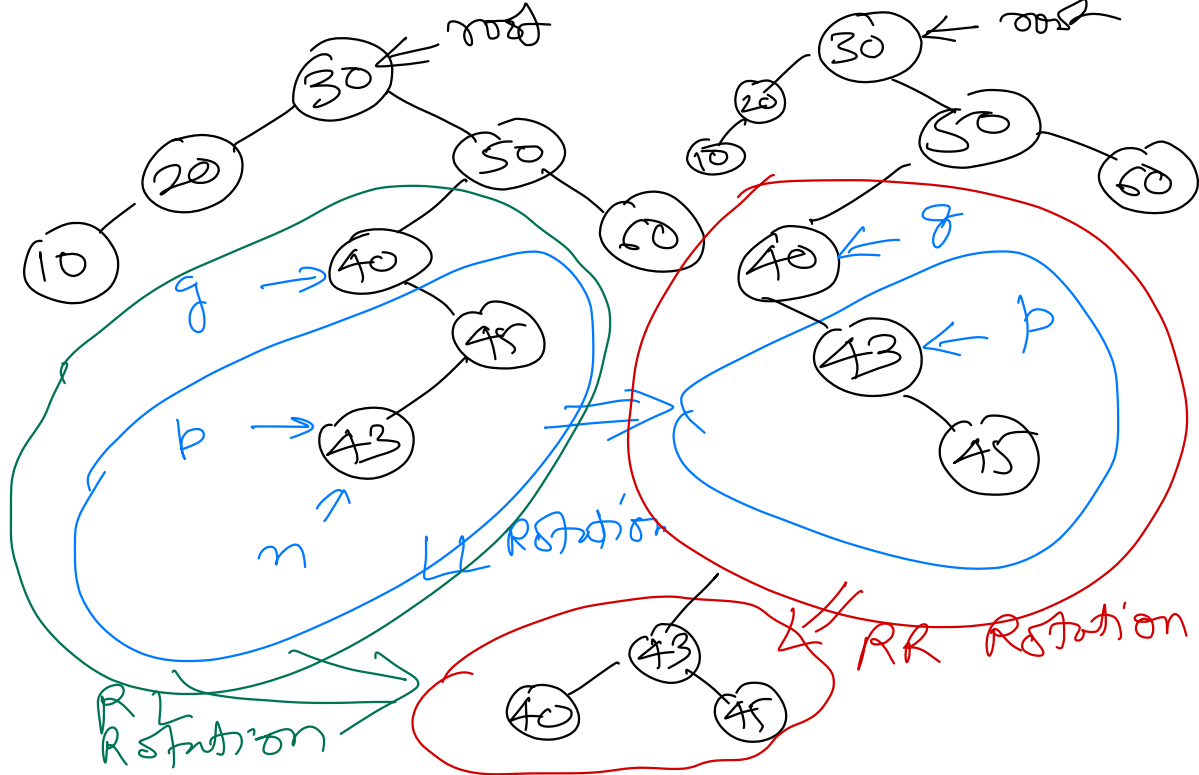
⇒ RR rotation around p.

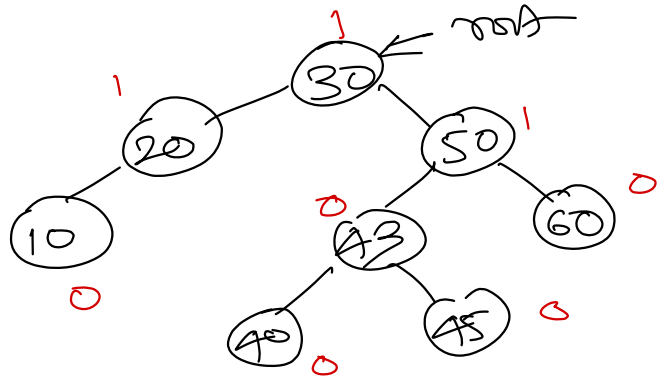
← RL (g, p)

LL (g's right child, p)
RR (g, p)

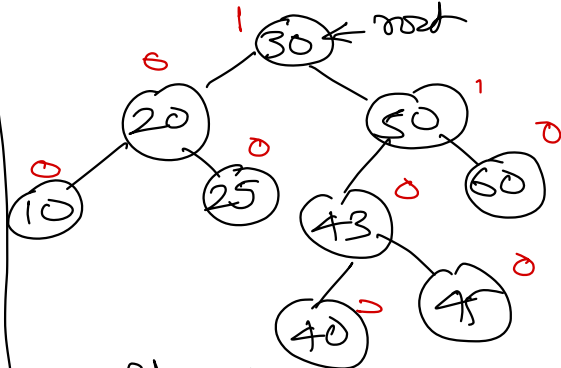
← LR (g, p)

RR (g's left child, p)
LL (g, p)

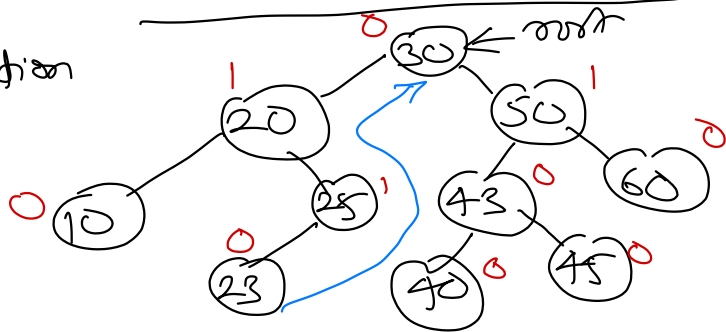




After inserting 43
 & rebalancing tree
 by doing RL rotation



After inserting 25



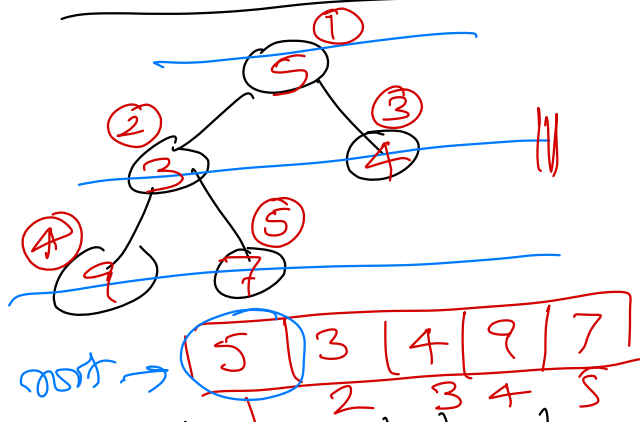
Jan Feb Mar Apr May June Jul Aug

Sep Oct Nov Dec

→ Ex: Babel AVL Tree to store name of months

Apr < Aug < Dec < Feb < ...

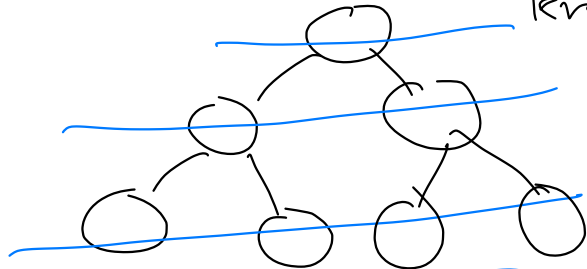
Complete tree



Full / Complete tree
then they can be
stored in an array.

Full tree

Every node has
2 children other than
leaves → on every
level.



Left i^{th} node
of tree
right
 $2 \times i$ $2 \times (i+1)$

