

## # AVL Tree:

BST: T1 => 30, 40, 10, 50, 20, 5, 35

BST: T2 => 50, 40, 35, 30, 20, 10, 5

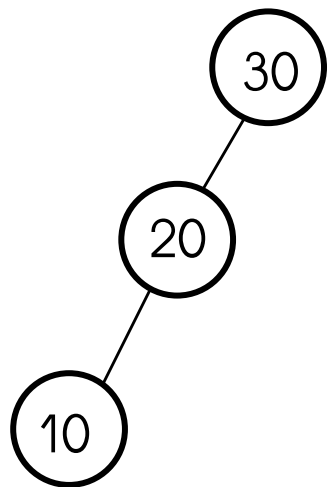
key elements are same in both T1 & T2 but we are adding ele's in different order, and after creation of BST height of T1 < height of T2.

---

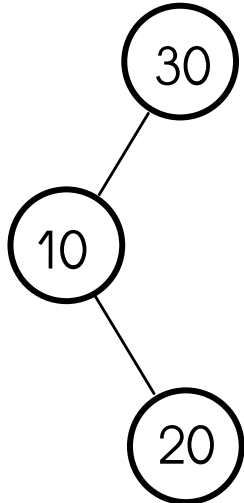
- if "n" no. of ele's are there we can have  $n!$  no. of orders/ways by which we can add ele's into the BST.

- e.g. if we have 3 ele's, so we can insert ele's by  $3! = 6$  possible orderes in which we can add ele's into the BST.

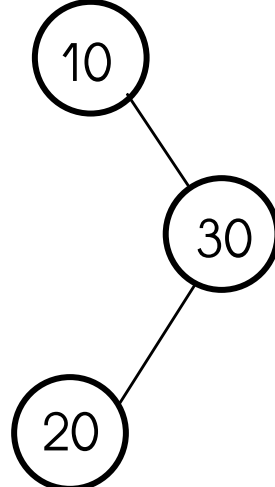
keys: 30, 20, 10



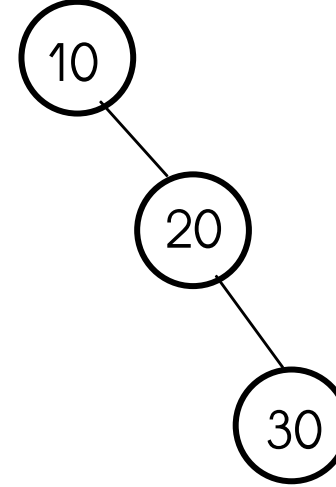
keys: 30,10,20



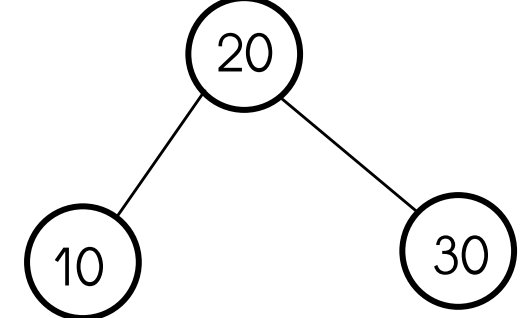
keys: 10,30,20



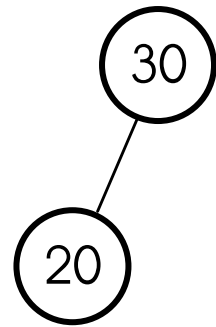
keys: 10,20,30



keys: 20,10,30 & 20,30,20

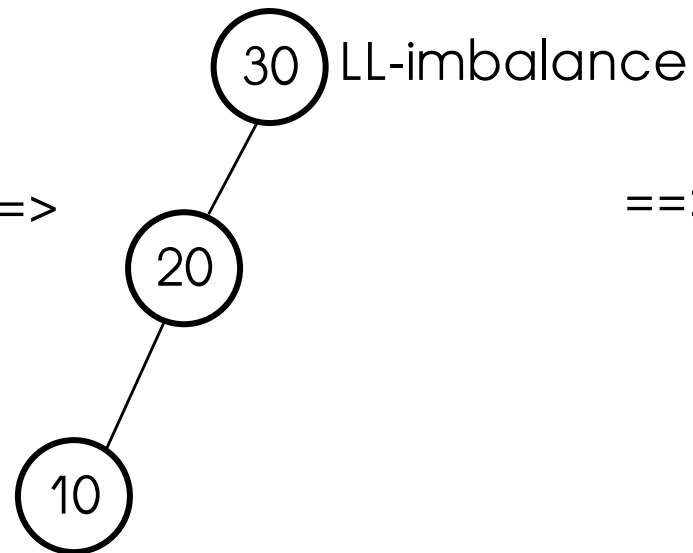


initially:



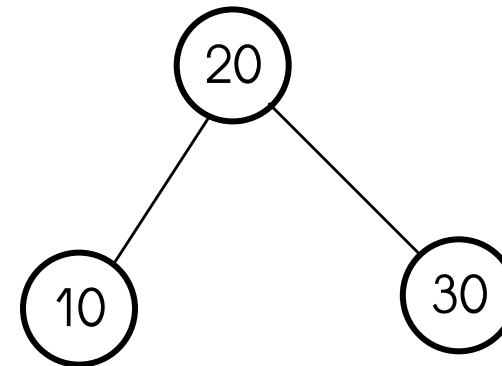
==>

insert: 10

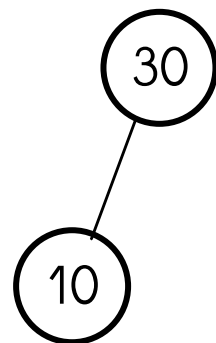


==>

After - LL Rotation

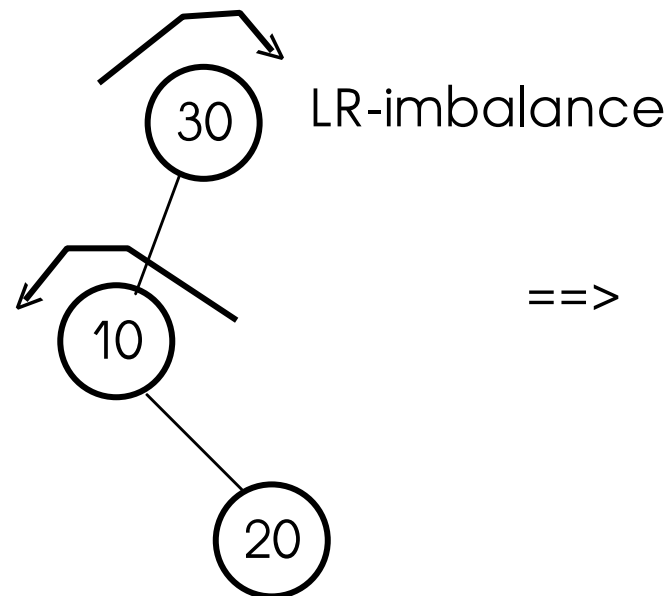


initially:



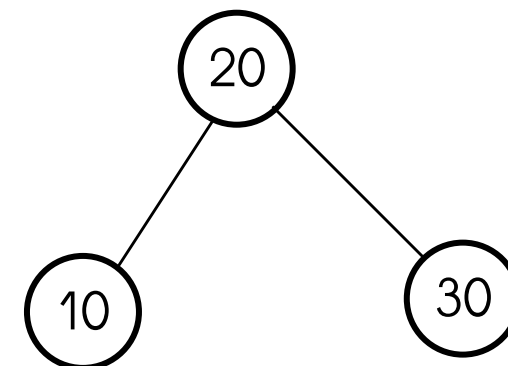
==>

insert: 20

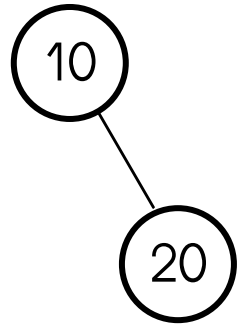


==>

After - LR Rotation

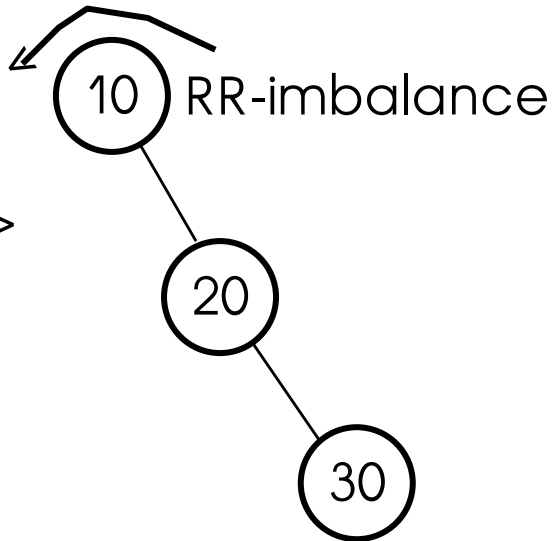


initially:



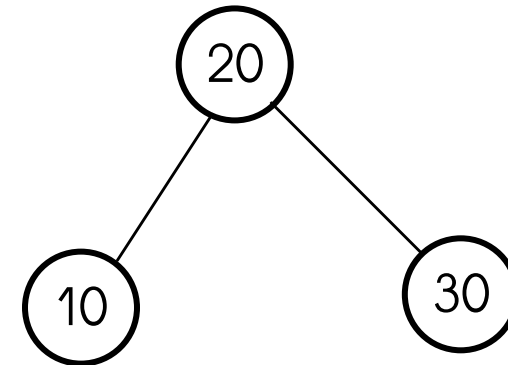
insert: 30

==>

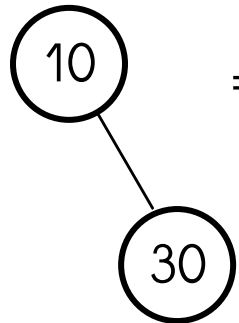


==>

After - RR Rotation

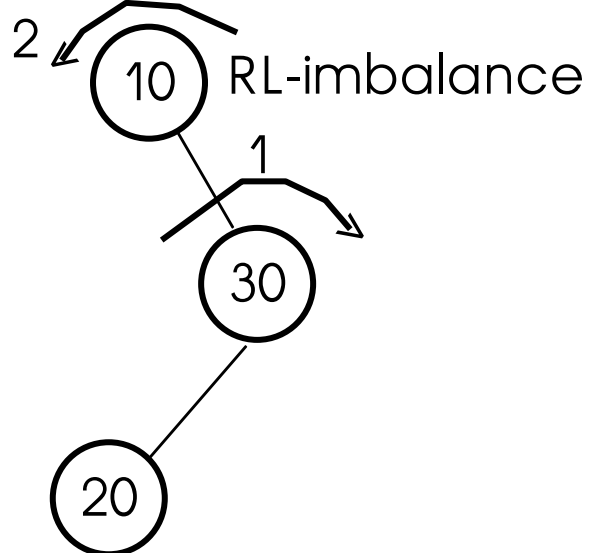


initially:



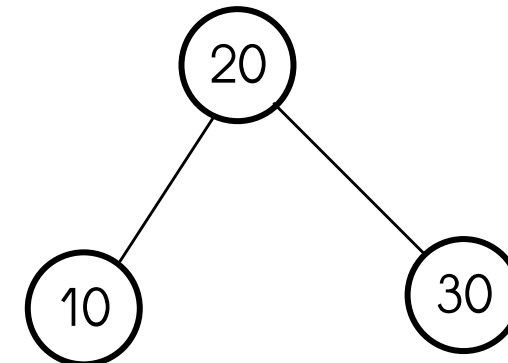
==>

insert: 20



==>

After - RL Rotation



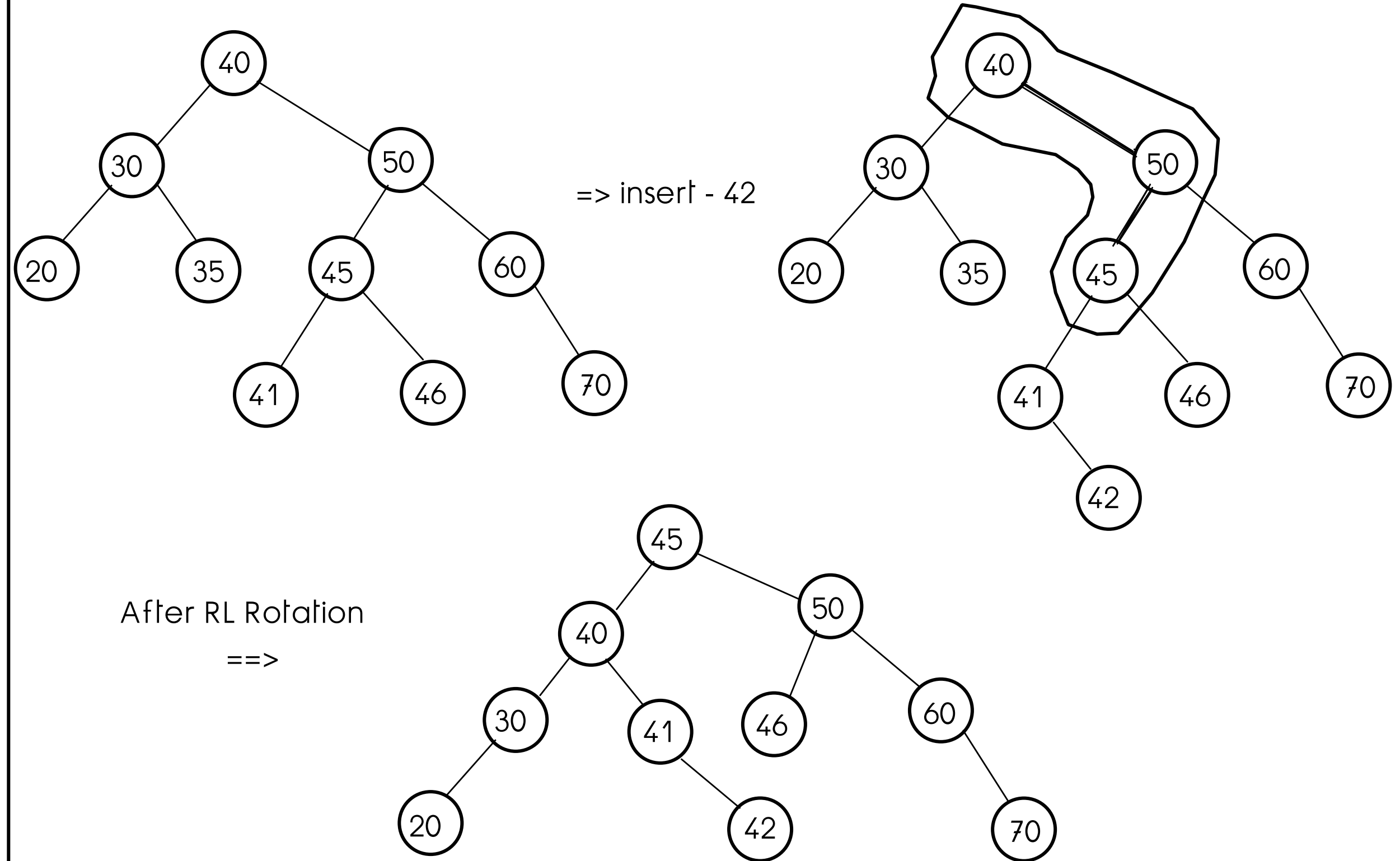
AVL Tree - Example:

RLLR imbalance => RL imbalance

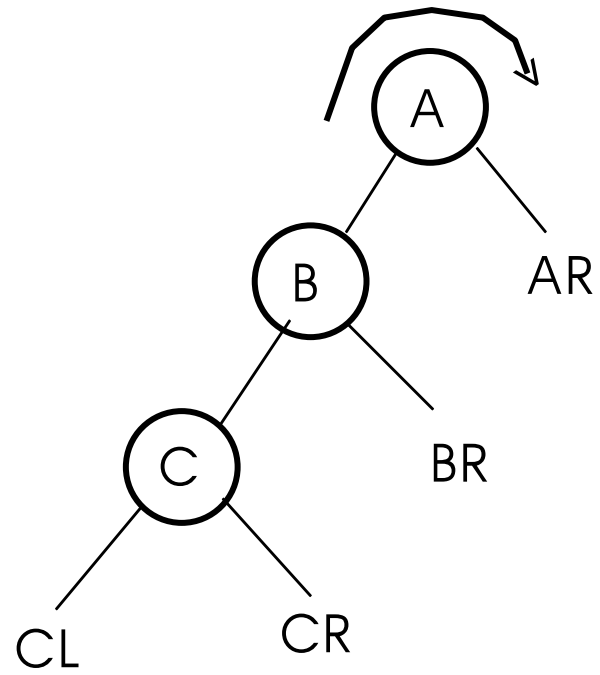
=> insert - 42

After RL Rotation

=>

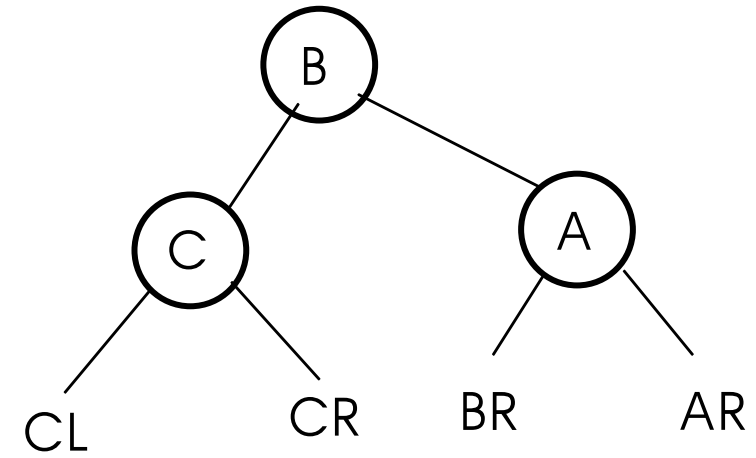


LL-imbalance

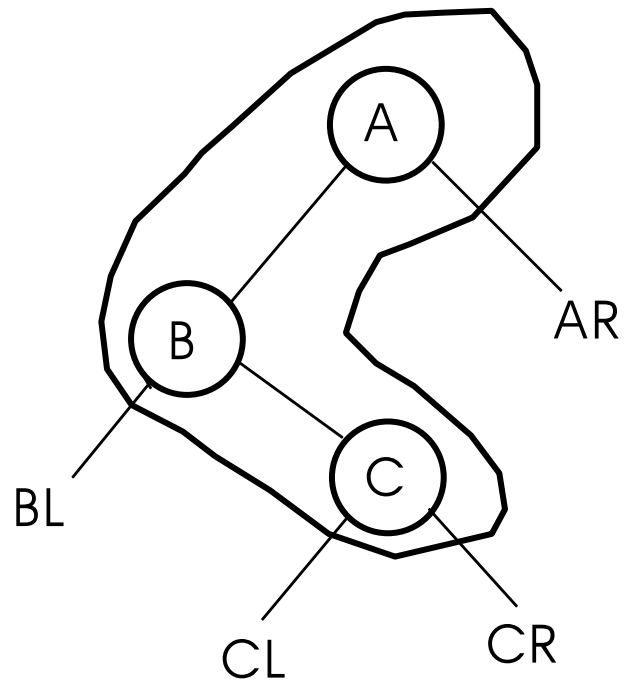


=>

LL Rotation



LR-imbalance



=>

LR Rotation

