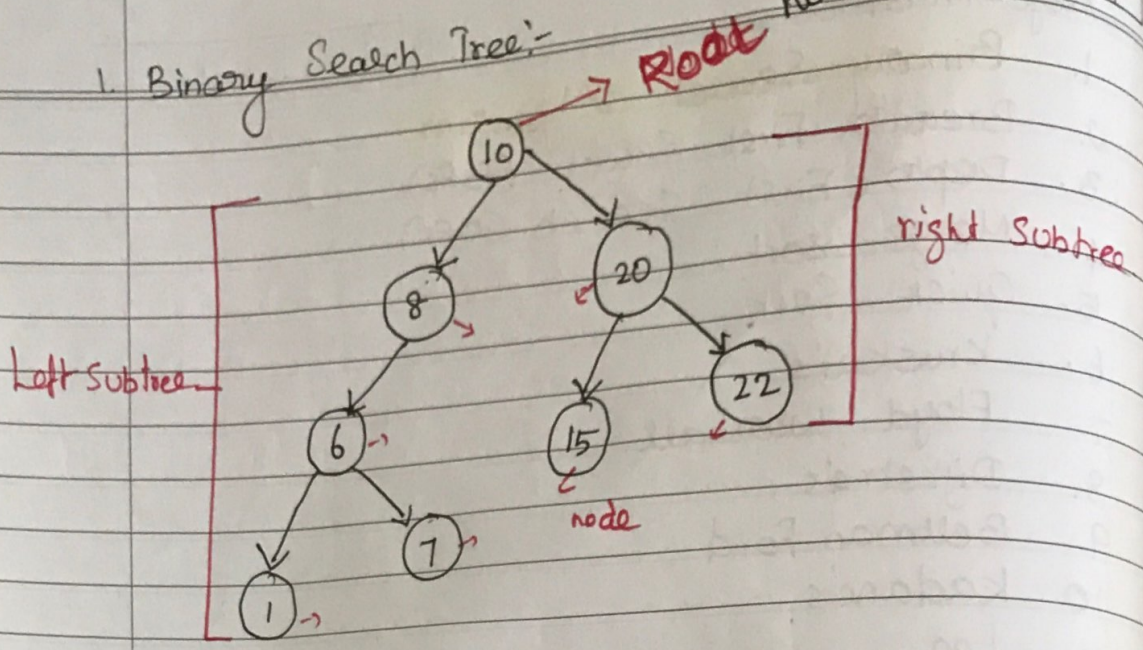
DATE

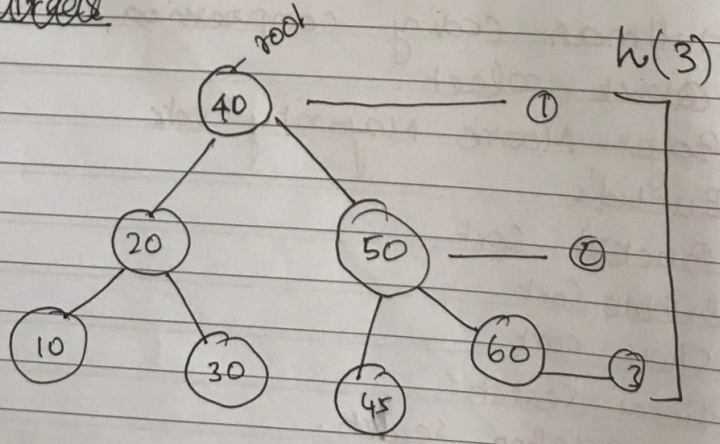
1. Binary Search Tree:-



Left subtree < Root node
right subtree > Root node.

Binary search trees used in many search applications where data is constantly entering/leaving. Such as the map and set objects.

Big O of Bst is O(n) (worst case)
(insertion, deletion and searching).
O(h) → depends on the height.



So worst O(n)
best O(log n)

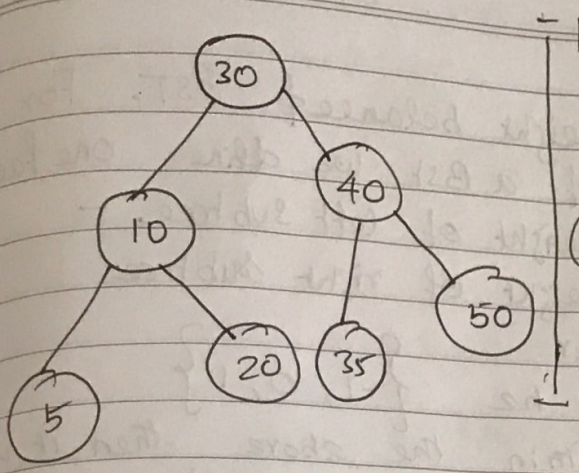
No duplicates in Bst

classmate

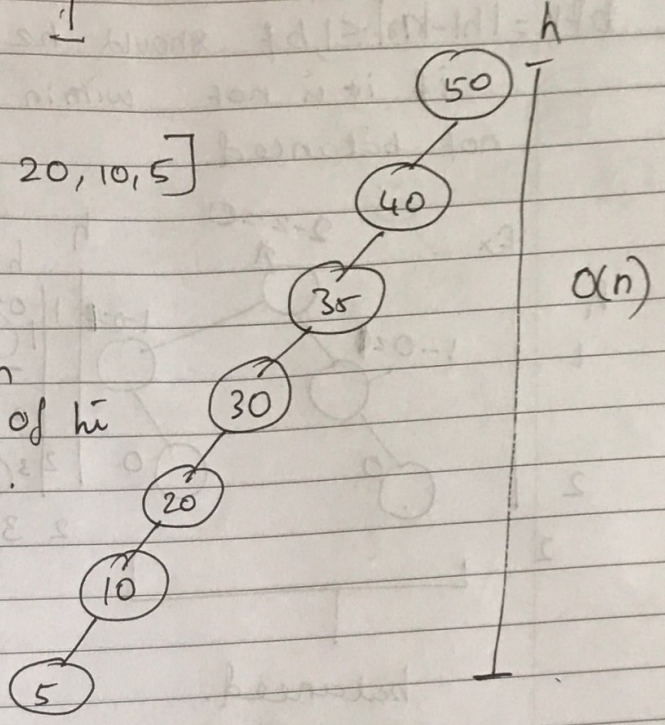
PAGE

Ex: Keys [30, 40, 10, 50, 20, 5, 35]

DATE



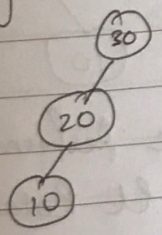
[50, 40, 35, 30, 20, 10, 5]



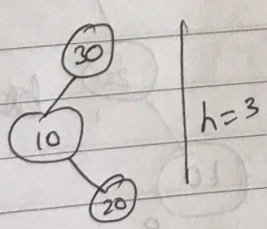
Big(O) Based on the height of the trees.

Orders of Tree. = 3! (3X2X1) = 6 orders

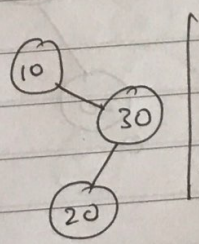
Key(30, 20, 10)



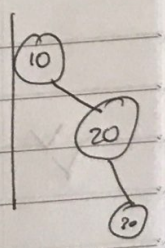
30, 10, 20,



10, 30, 20

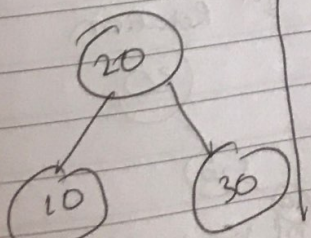


10, 20, 30



20, 10, 30

20, 30, 10



PAGE

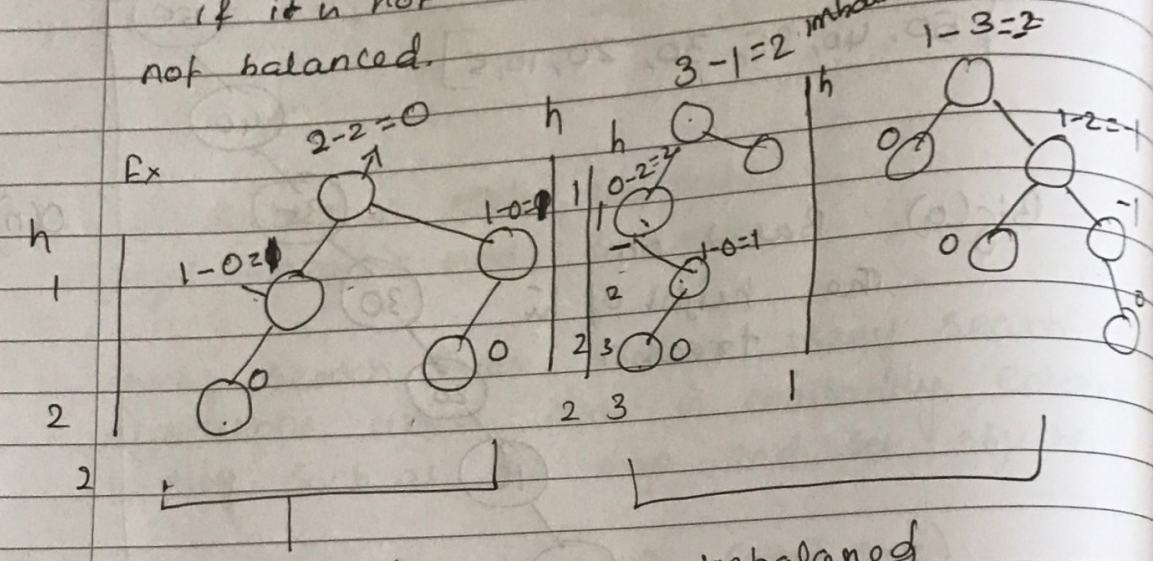
DATE

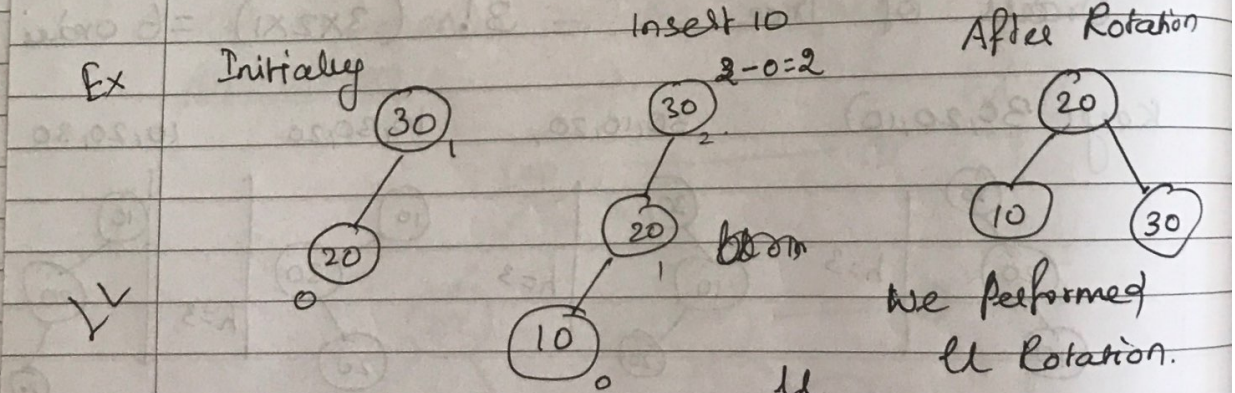
AVL Trees

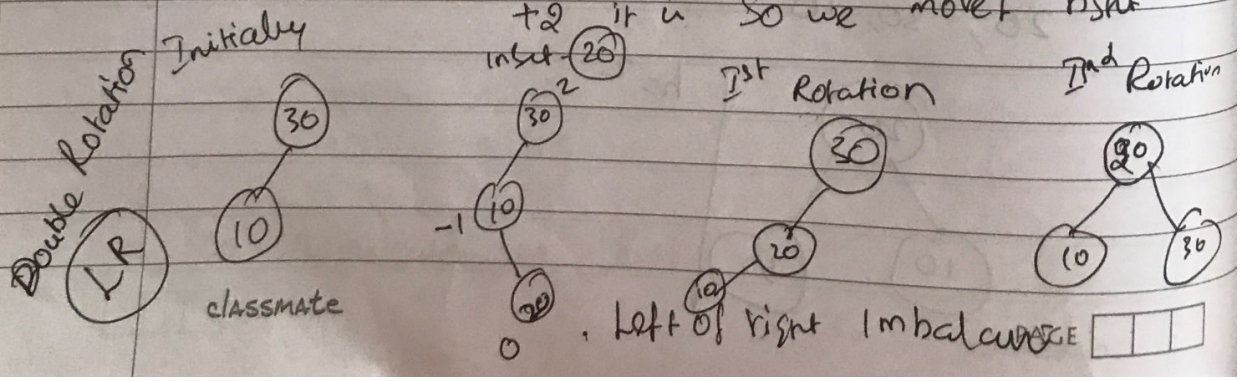
AVL tree is a height balanced BST. For balancing the height of a BST we define one factor balance factor = height of left subtree − height of right subtree

$bf = h_l - h_r$

$|bf| = |h_l - h_r| \leq 1$, bf should be $\{-1, 0, 1\}$ if it is not within the above then it is not balanced.

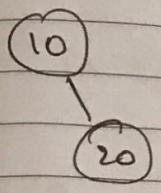




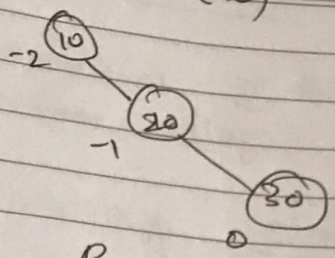


DATE

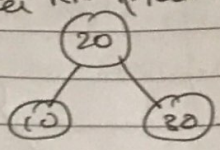
Initially



Insert (30)

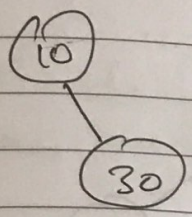


After RR Rotation

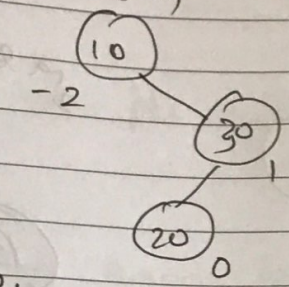


Right of right Imbalance

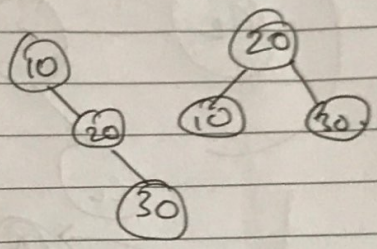
Initial



Insert (20)



RL Rotation

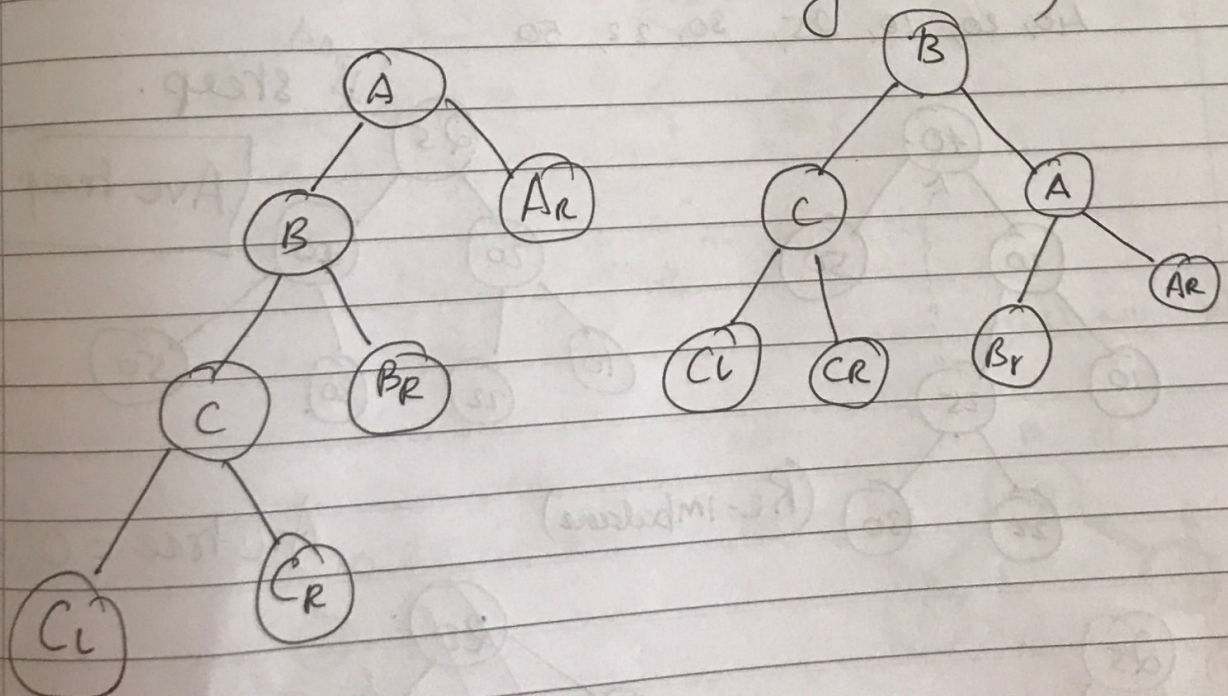


RL Imbalance

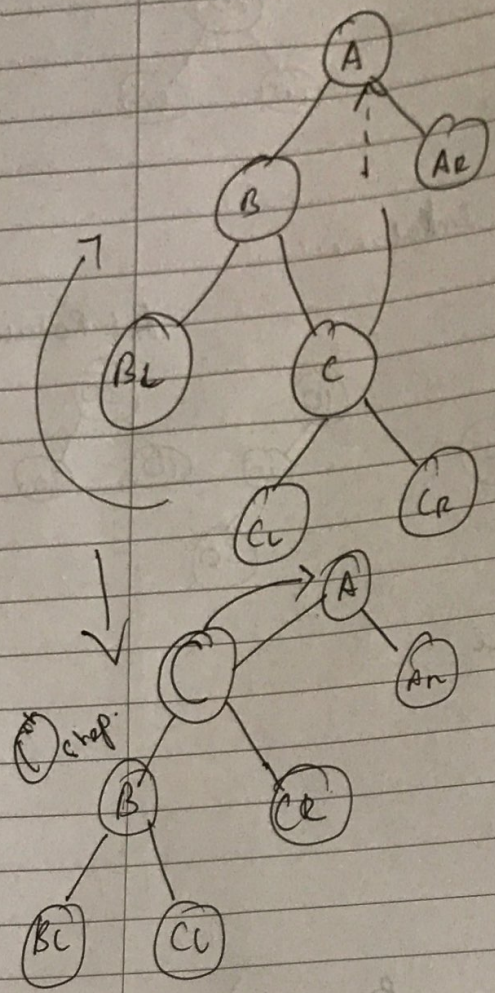
LL
RR
Single Rotation.

LR
RL
Double Rotation

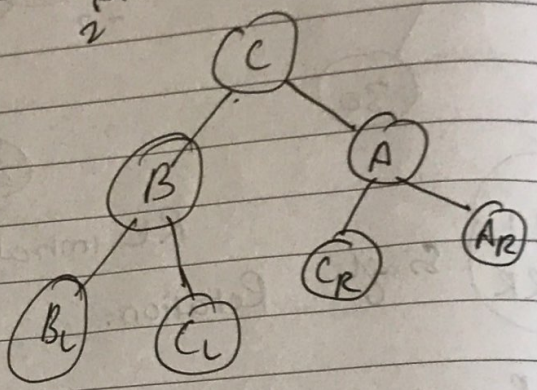
LL - Rotation (with many nodes)



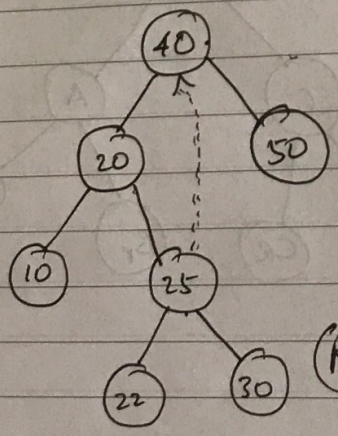
LR Rotation



2nd step



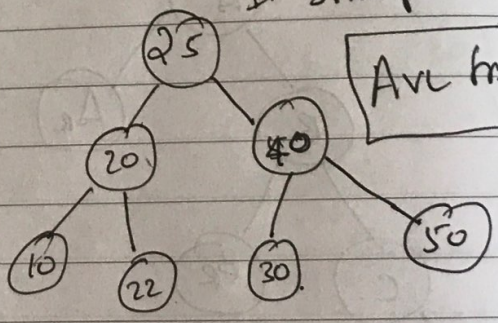
40, 20, 10, 25, 30, 22, 50



(RL-Imbalance)

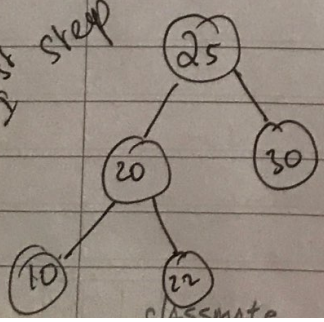
II nd step.

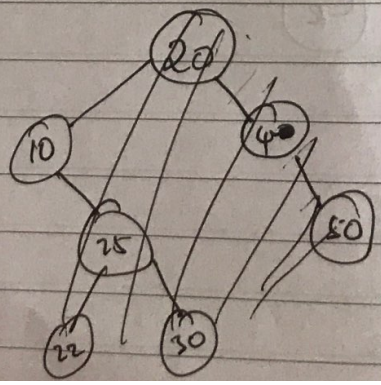
Avl tree



Avl tree = O(lg

I st step





Red-Black is same as AVL Tree.

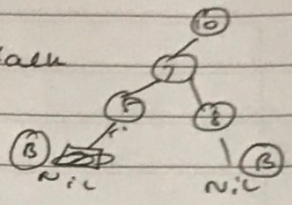
DATE

Based on Height O(n) or O(log n).

in RB · maximum 2 Rotations are required. conform O(log n)

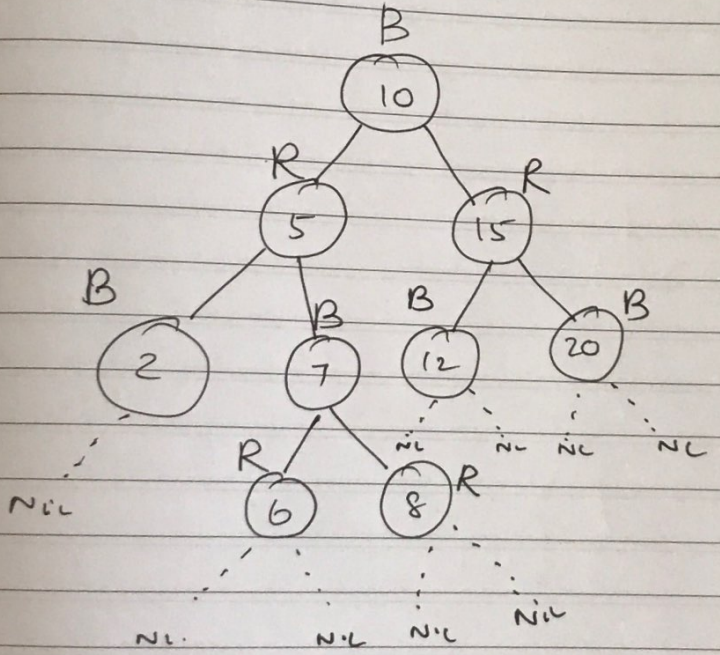
★ Root is always Black (head node)

★ Every leaf which is Nil is Black



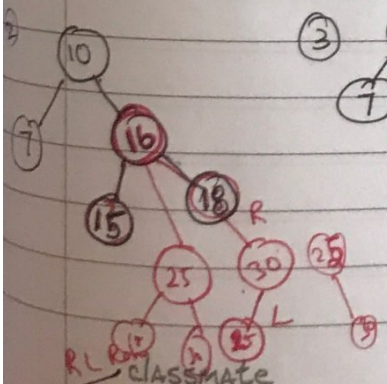
★ If node is red then its children are black

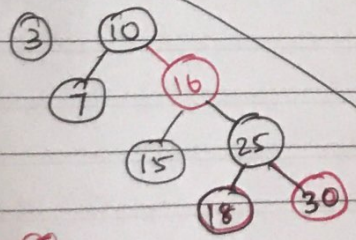
★ Every path from a node to any of its descendant Nil node has same no. of black nodes.

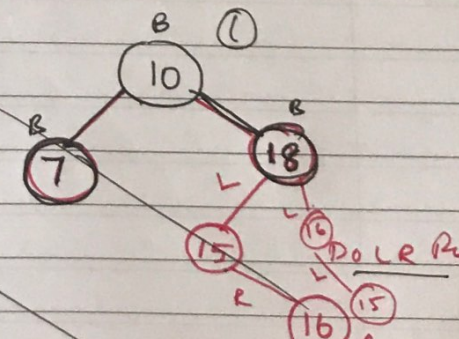


all Nil is Black

Ex 10, 18, 7, 15, 16, 30, 25, 40, 60, 2, 1, 70







Learn insertion of Red-Black tree.

Confusing.

PAGE