"AVL Tree" :-

12/oct/2017

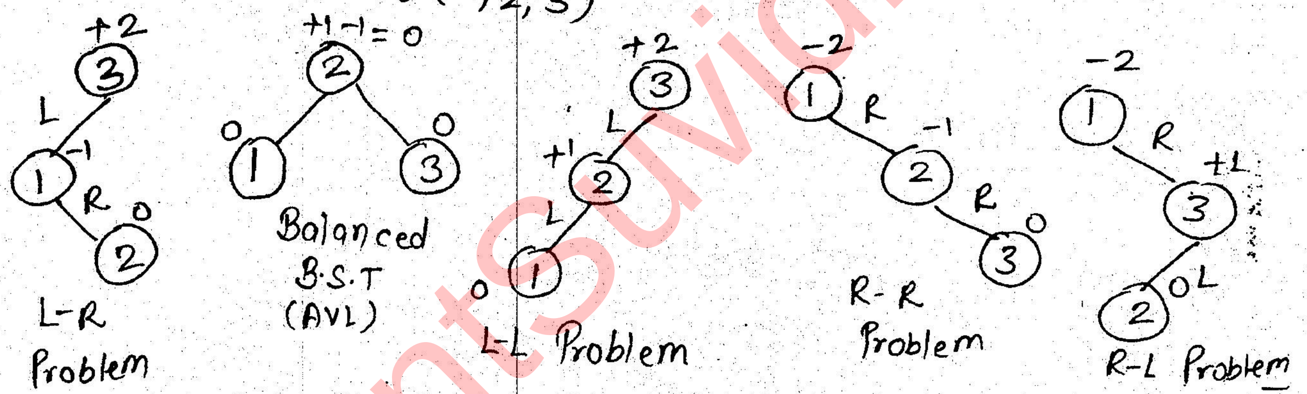
(Adelson, Valsky, Londys)
(or)
Balanced B.S.T.
(or)
Height Balanced Tree (Height of Left subtree && Right subtree equal)

:- At Every Node balancing factor (+1, 0, -1)

"Balancing factor" = Height (LST) − Height (RST)

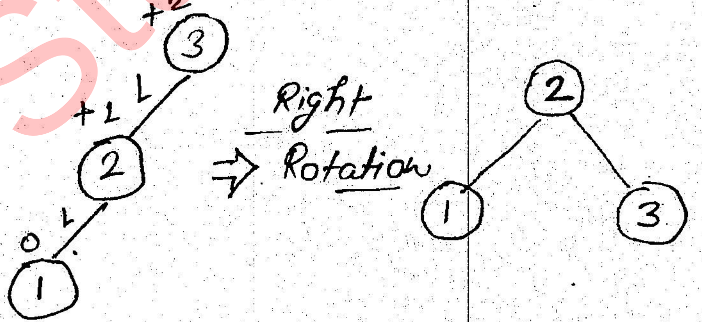
Ex: ① :-

n = 3 (1, 2, 3) BST's :-



"Rotations" :-

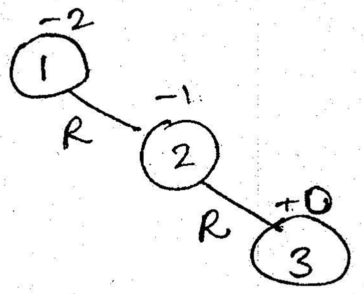
i) L-L Problem :-



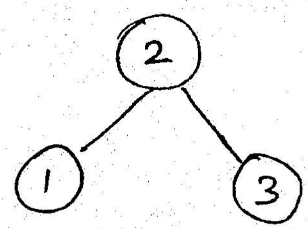
2 → rc = 3
3 → LC = Null } ⟹ O(1)

Parent of 2 came to Right side, "Right Rotation"

ii) R-R Problem :-

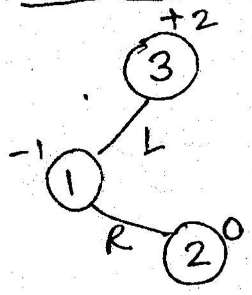


Left Rotation ⇒

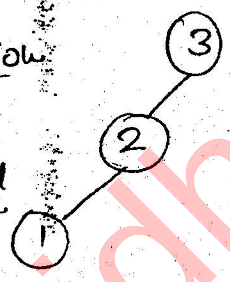


2 → LC = 1
1 → RC = Null
⇒ O(1)
Parent of 2 came to left, "Left Rotation"

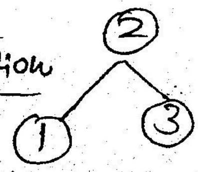
iii) L-R Problem :-



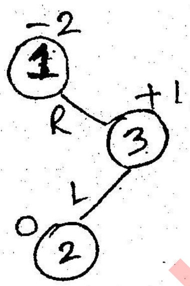
⇒ Left Rotation
2 → LC = 1
1 → RC = Null
⇒ O(1)



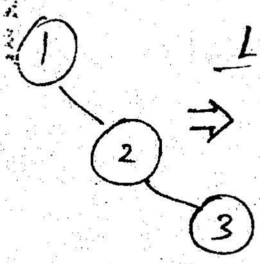
⇒ Right Rotation
2 → RC = 3
3 → LC = Null
⇒ O(1)



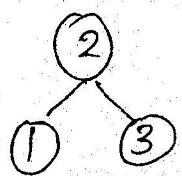
iv) R-L Rotation :-



⇒ Right Rotation
2 → rc = 3
3 → LC = Null

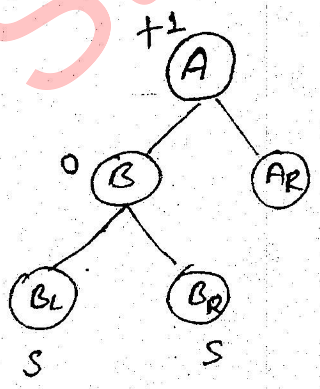


⇒ Left Rotation
2 → LC = 1
1 → RC = Null



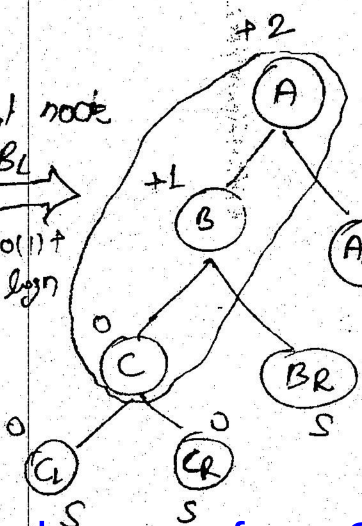
"Insertion" :-

L-L Problem :-

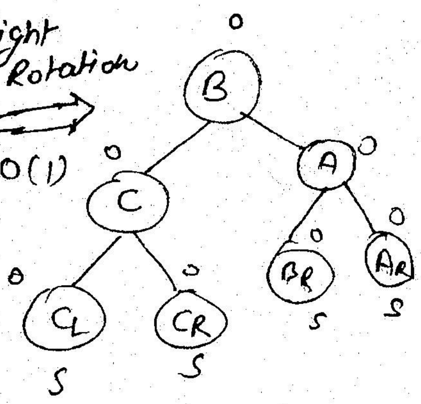


close to L-L Problem

Insert node into BL ⟹
log(n) + O(1) + O(1) + log n



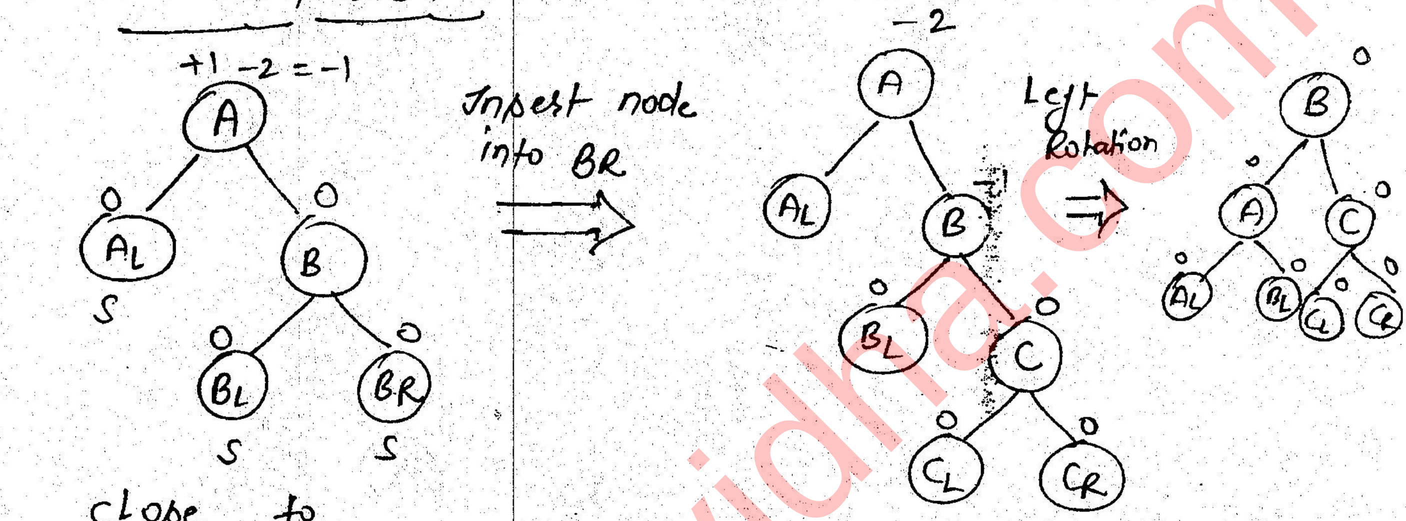
Right Rotation ⟹ O(1)



Time :- $\underbrace{\log n}_{\text{for inserting Place}} + \underbrace{O(1)}_{\text{creation Node}} + \underbrace{O(1)}_{\text{linking}} + \underbrace{O(1)}_{\text{Rotation}} + \underbrace{\log(n)}_{\text{varifying back}}$

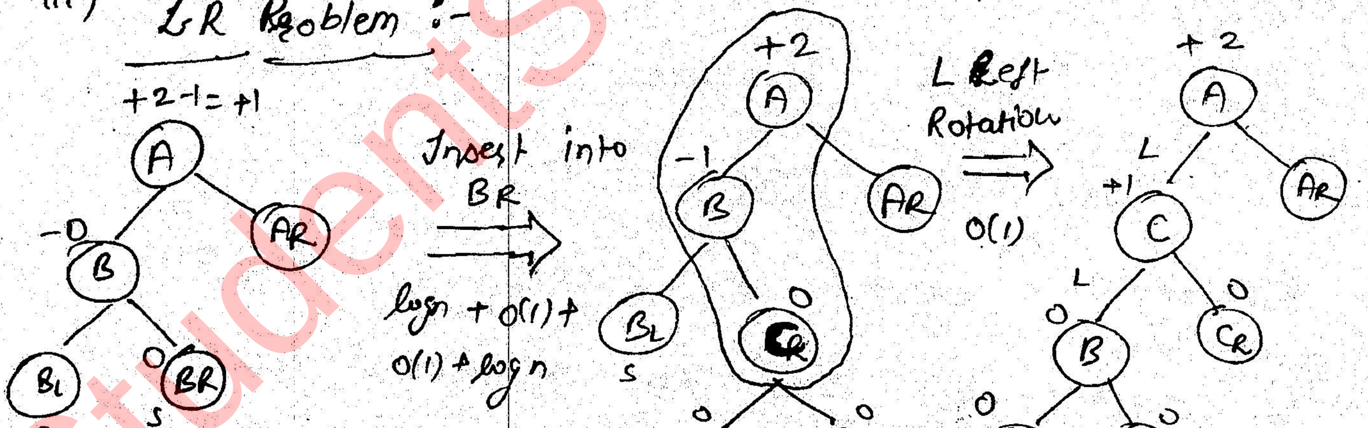
$\Rightarrow$ $O(\log n)$

ii) R-R Problem :-



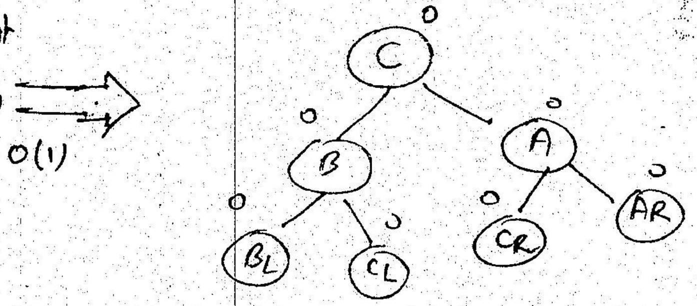
close to R-R Problem

iii) LR Problem :-

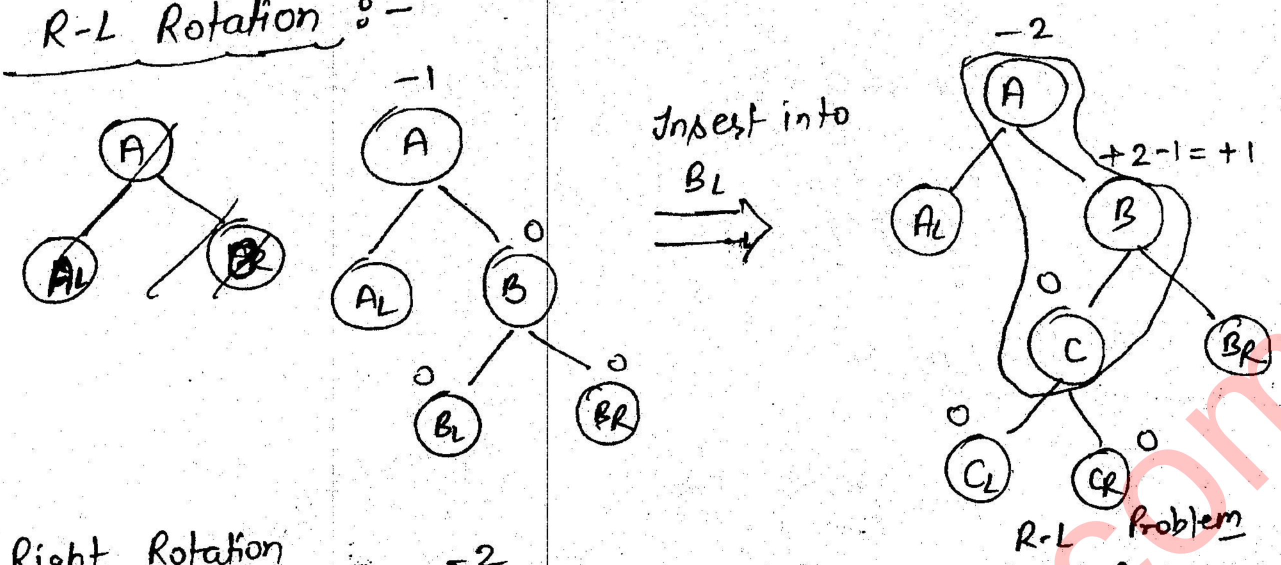


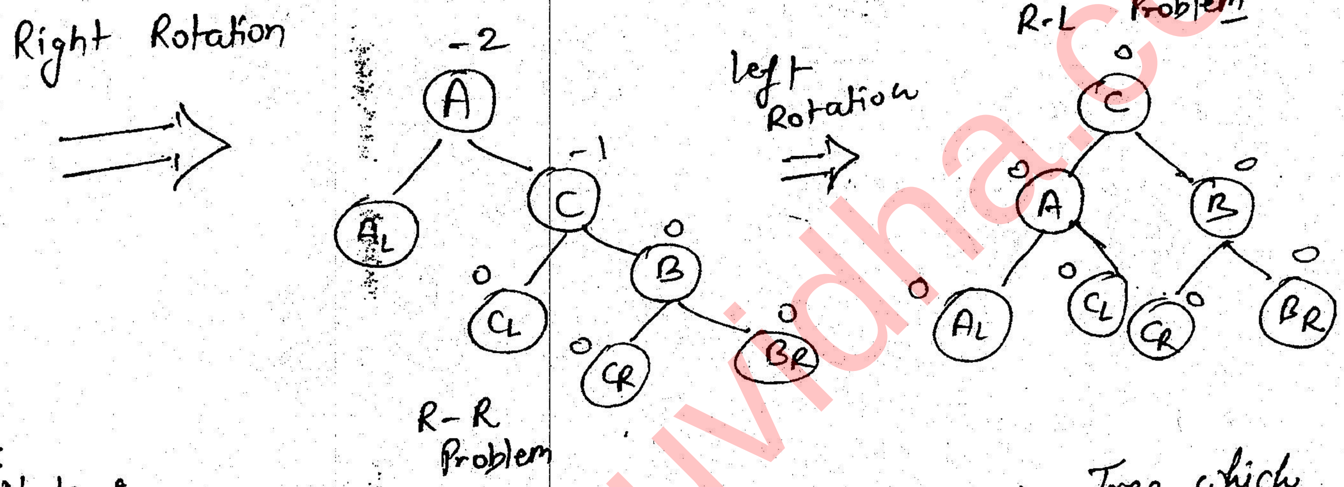
close to L-R Problem

Right Rotation O(1)



1) R-L Rotation :-





*Note :-

i) Inserting a Node (element) into AVL Tree which already contain n- nodes will take O(logn) every case.

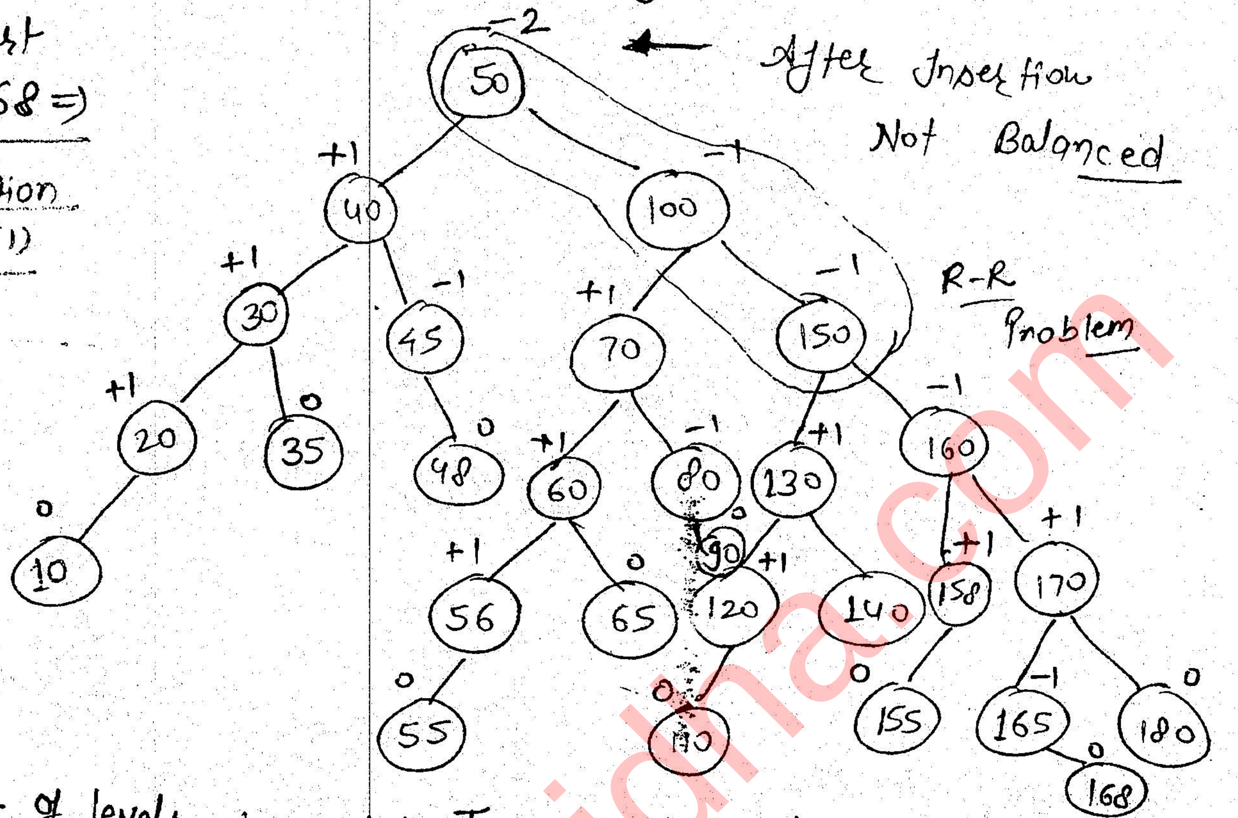
Inserting n- element ⇒ O(n.logn) [Every case]

ii) To create 'AVL Tree' for n element will take O(nlogn) [every case] but for BST $O(n^2)$ [worst case]

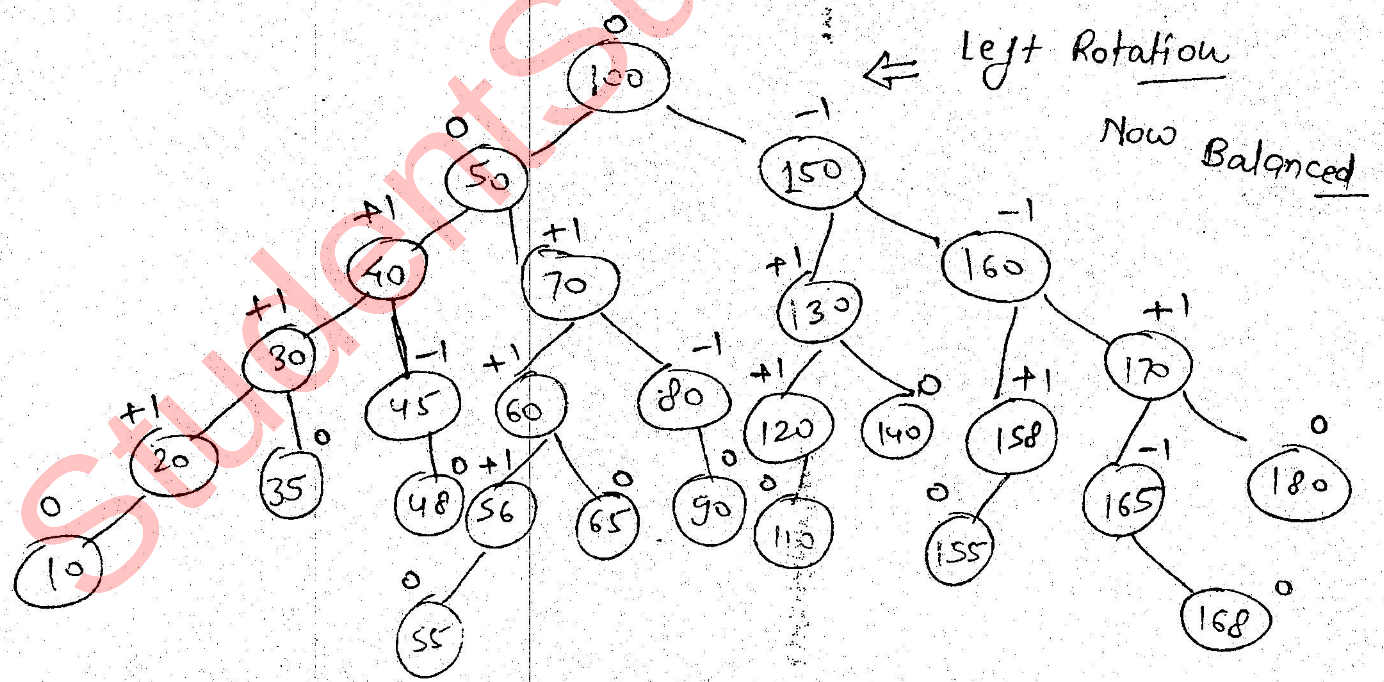
Que :- Consider the following AVL Tree :

Insert 168 ⇒

Insertion O(1)

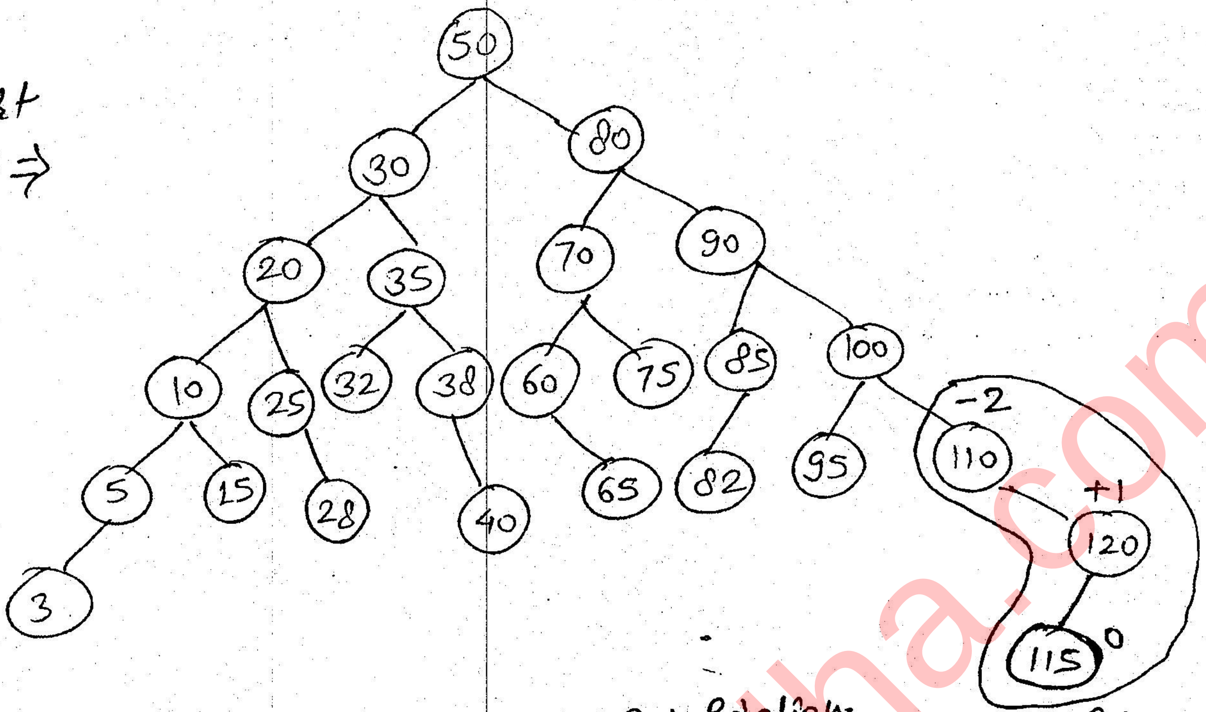


# of levels in AVL Tree with n nodes (max, min) = $\log_2 n$ [Asymptotically]

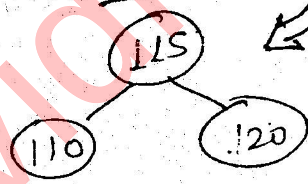


Que: Consider the following AVL Tree :

Insert 115 ⇒



R-L Rotation

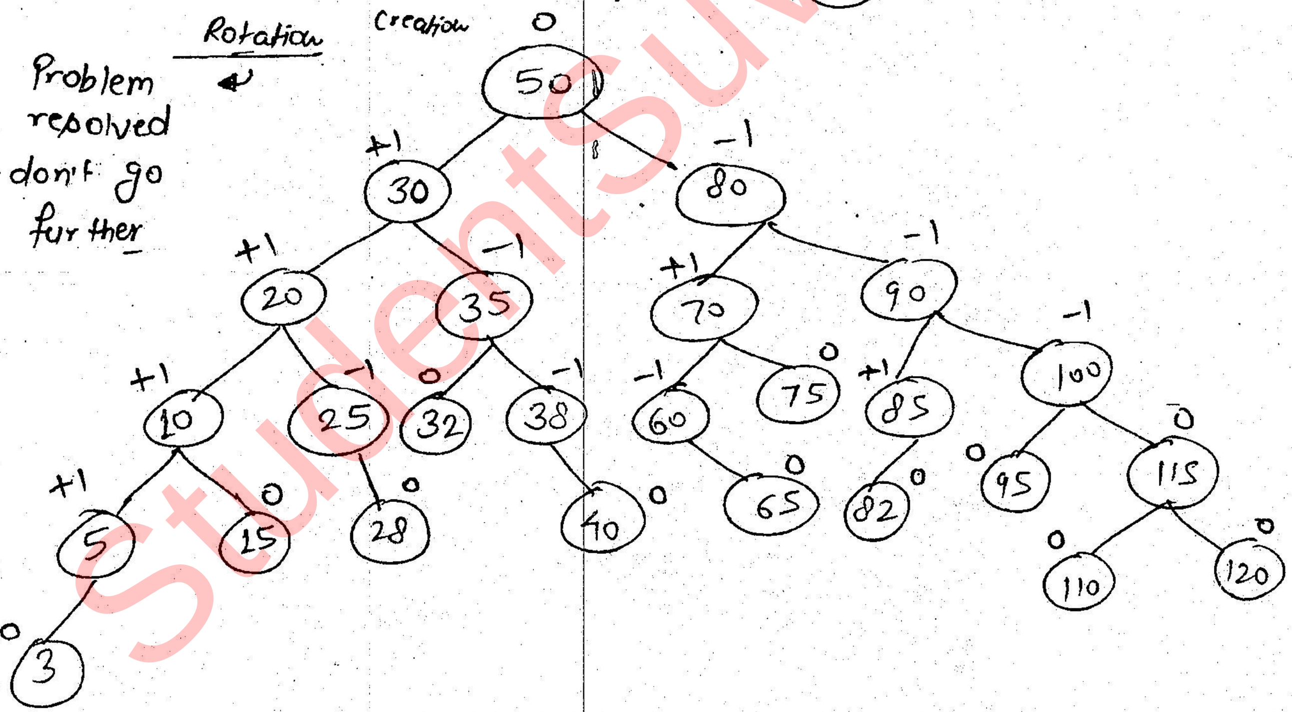


Time : $\underline{\log n}$ (finding insertion place) $+ O(1) + \underline{O(1)}$ (linking)

$+ \underline{O(1)}$ (Node creation)

Rotation

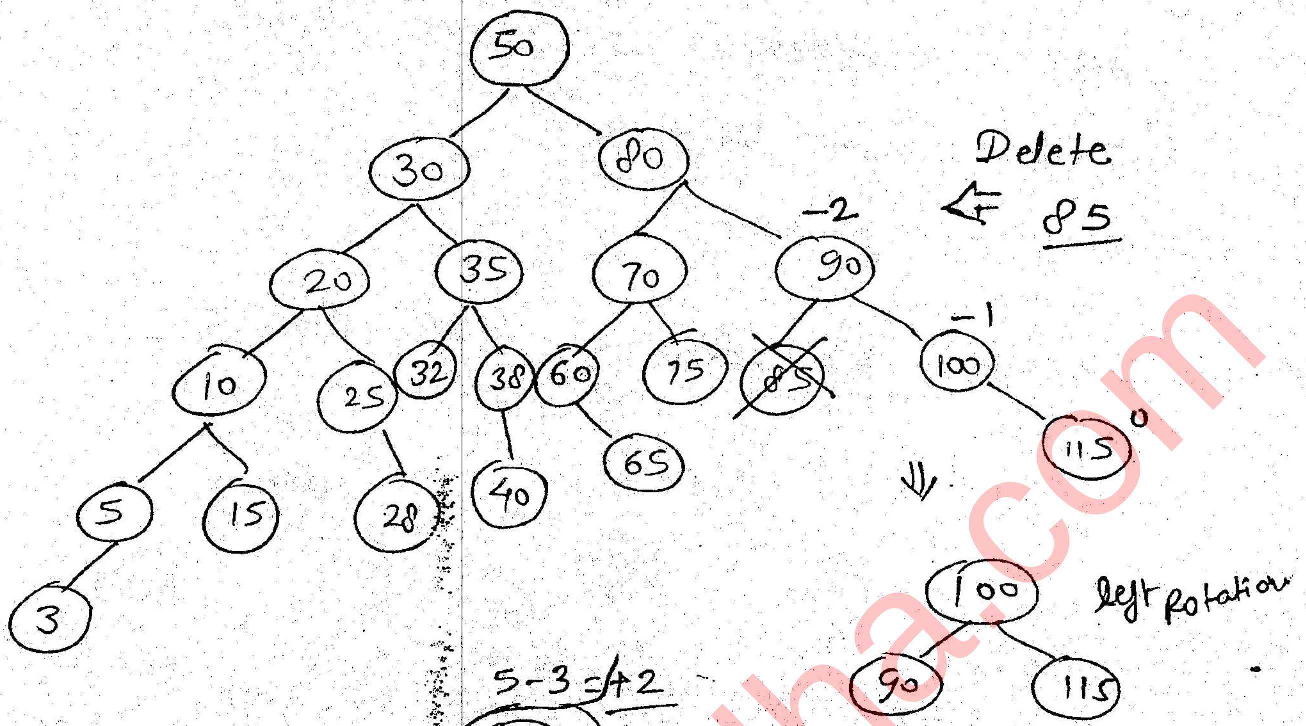
Problem resolved don't go further

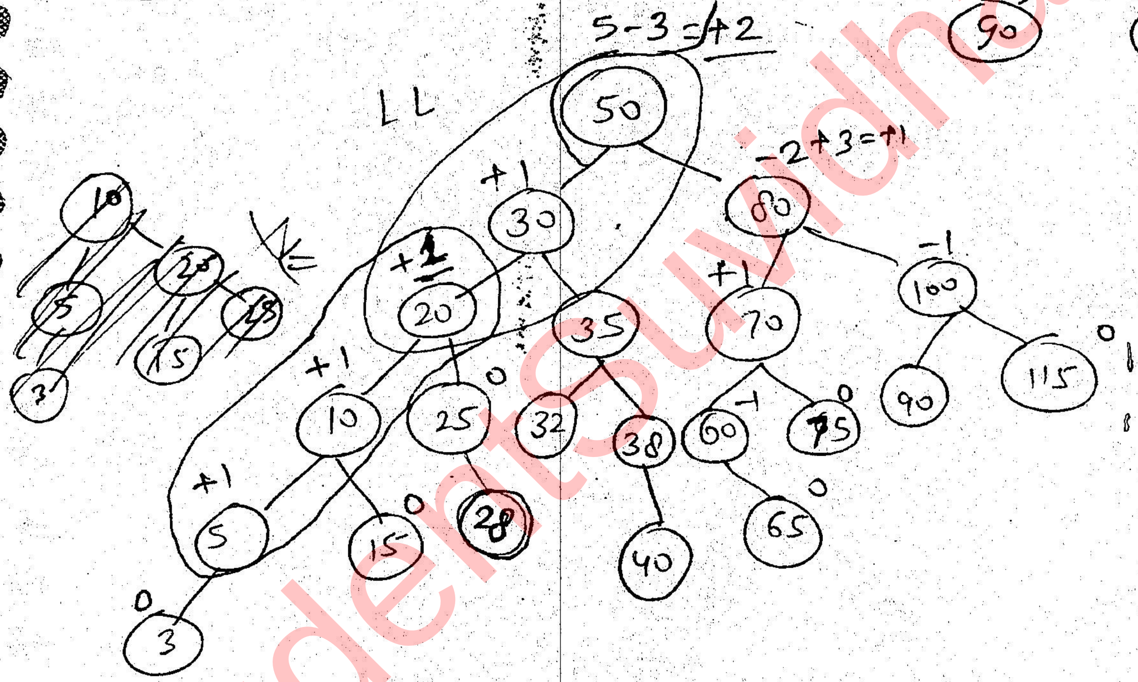


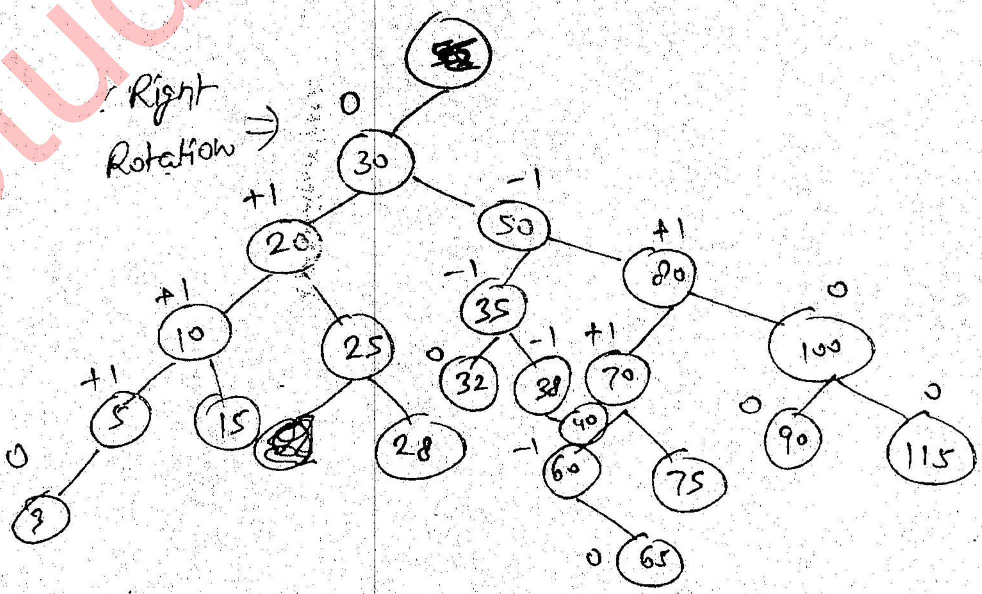
★ Note :- After Insertion while going back in that path maximum one problem can happened.

[max two Rotations]

Que :- Consider the following AVL Tree :



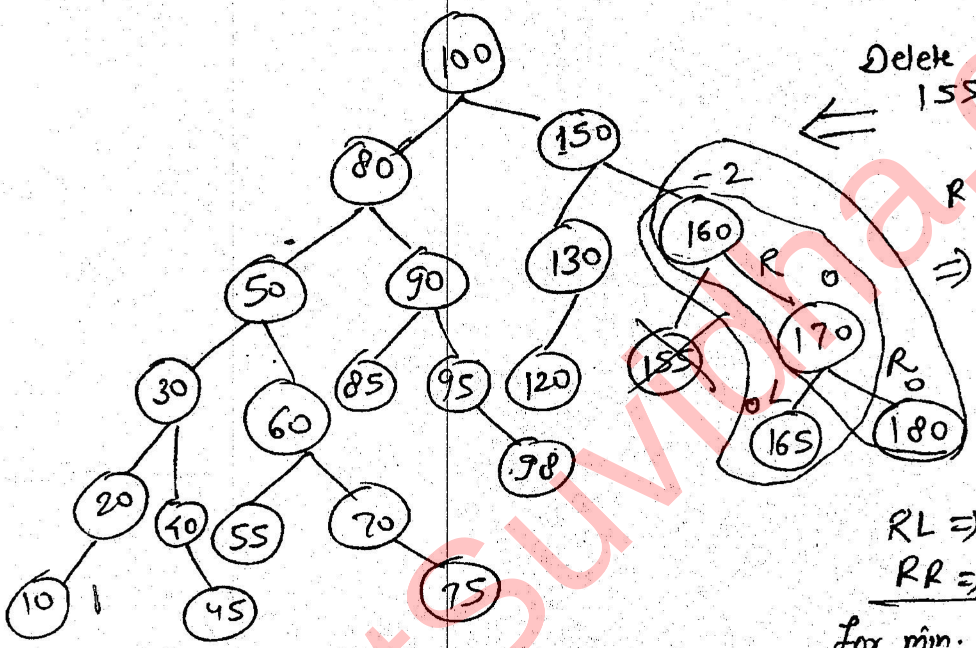




Note :- After deletion while going back maximum logn problems can come across [max. 2 logn rotation].

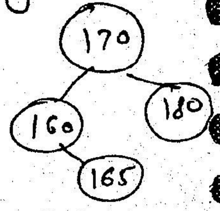
:- minimum, 0 problem.

Que :- Consider the following AVL Tree :-



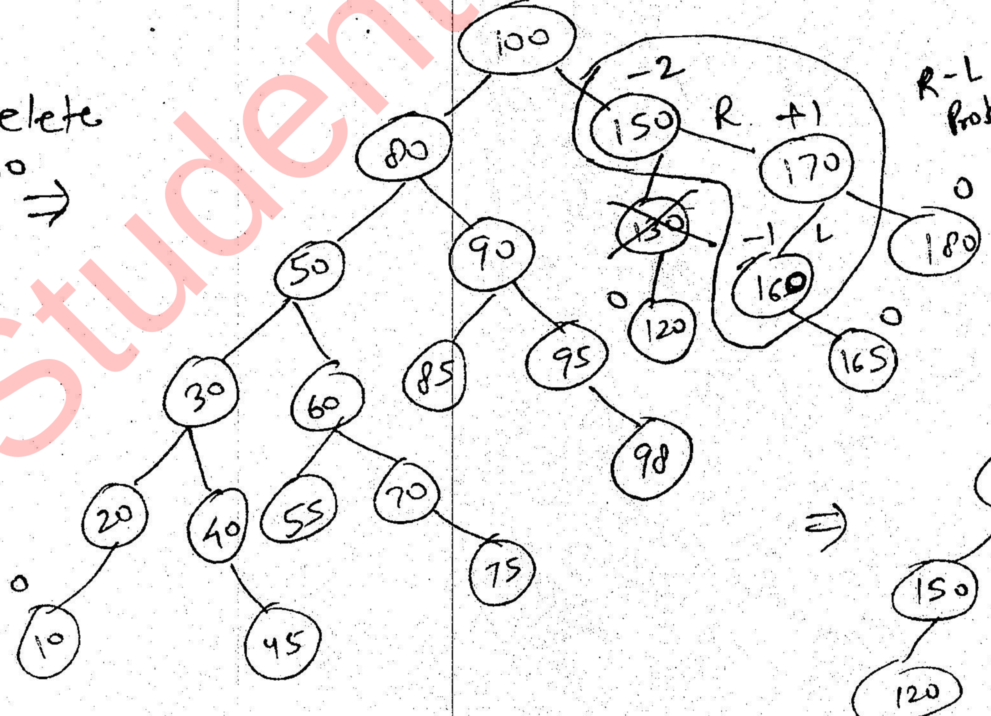
Delete 155

RR Problem so left Rotation ⇒



RL ⇒ 2 rotations
RR ⇒ 1 rotation
for min. rotation take RR

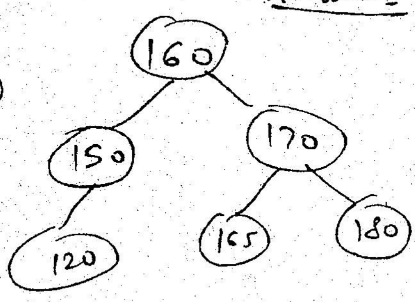
Delete 130 ⇒

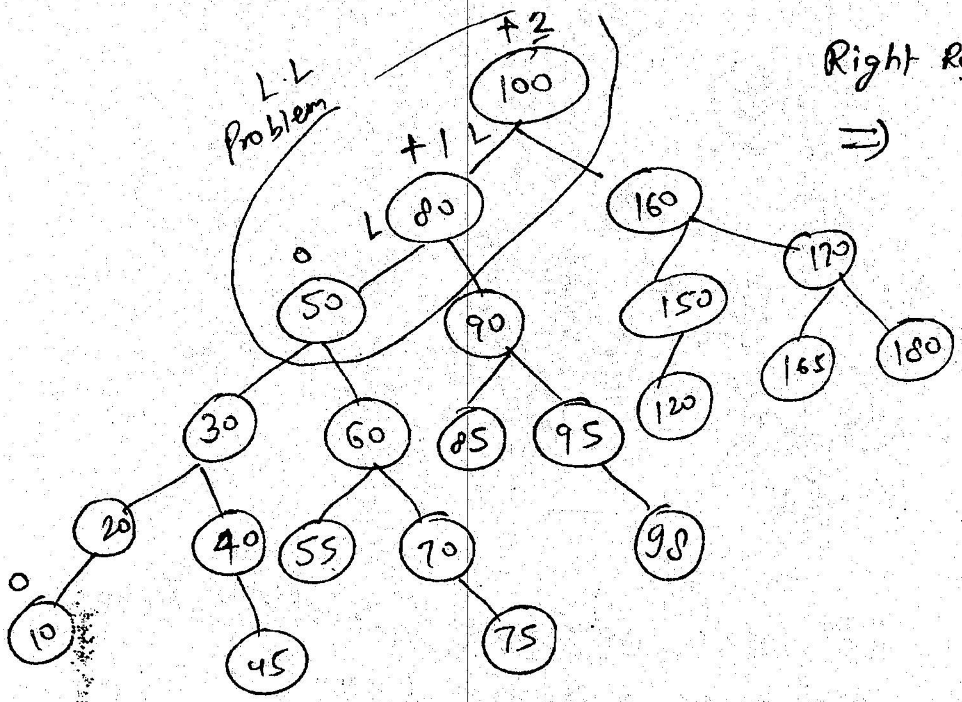


R-L Problem

⇓ Right Rotation then Left Rotation

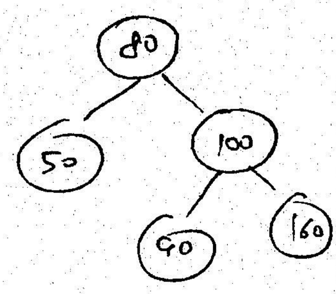
⇒



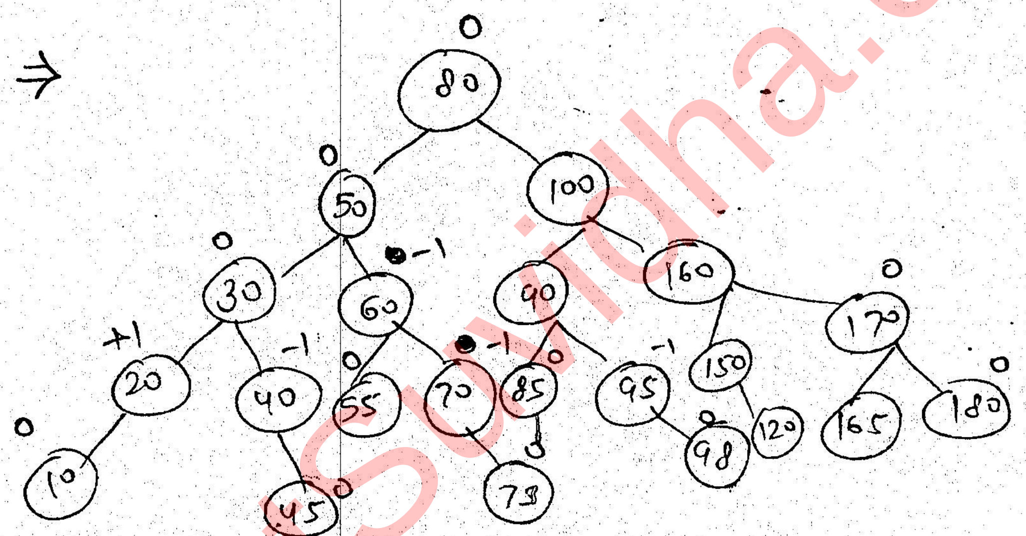


Right Rotation

$\Rightarrow$



AVL Tree $\rightarrow$



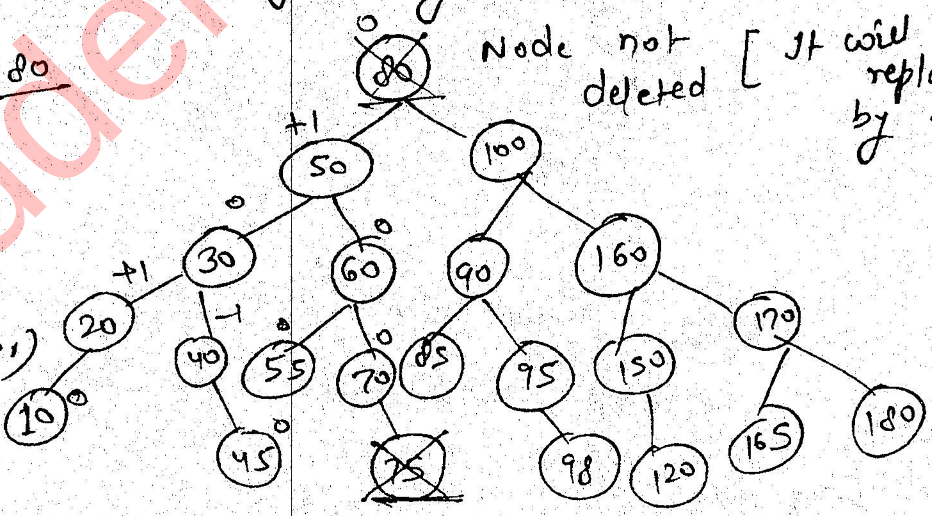
Que :- consider following AVL Tree :

Delete 80

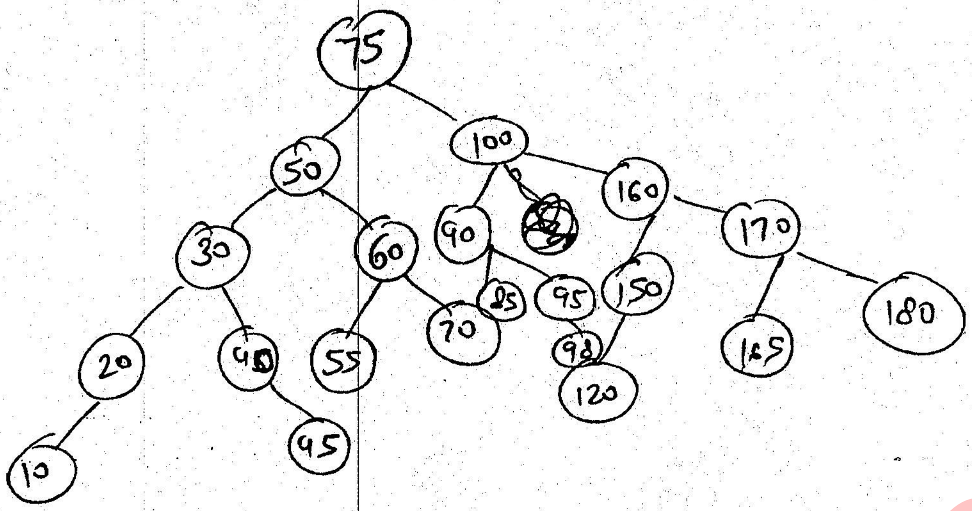
$\Rightarrow$

{ By Sir : Replace by Predecessor }

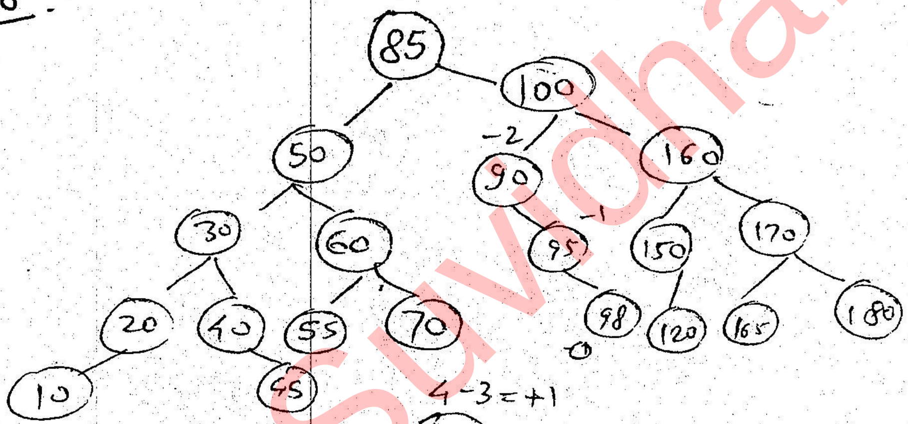
Node not deleted [It will be replaced by 75].

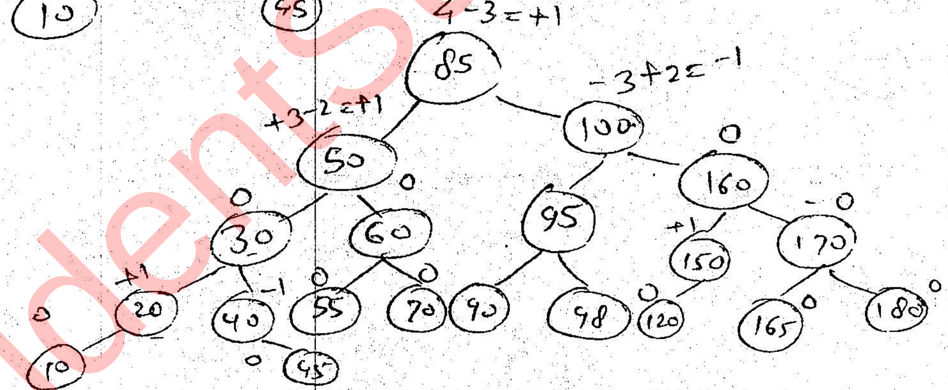


Predecessor Deleted



Delete 80 and replace by successor :- To 80 successor [smallest element on Right side = 85].

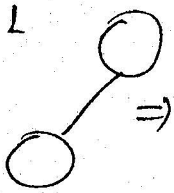




Deleting A Node from AVL Tree, which already contain n - element will take O (log n) time [Every case].

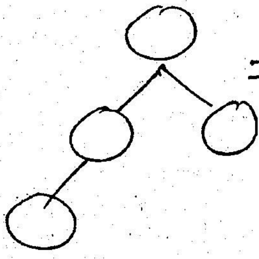
Que :- what will be maximum Height of AVL Tree with 20 Nodes.

h = 1



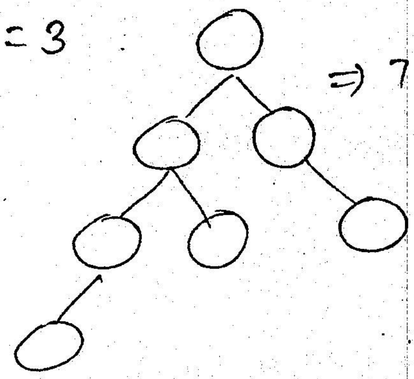
⇒ 2 element

h = 2



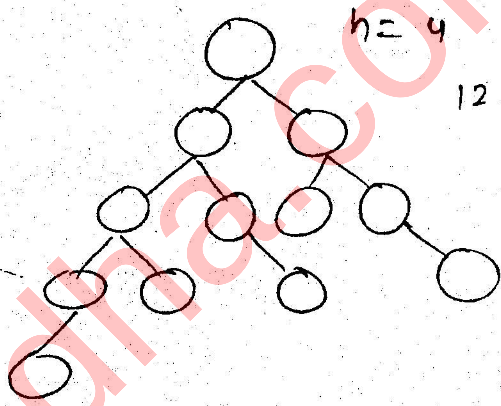
⇒ 4 nodes

h = 3



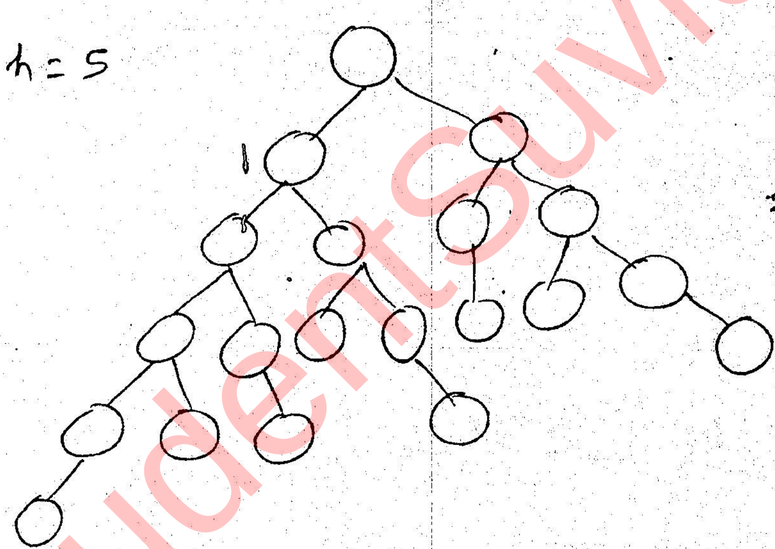
⇒ 7 nodes

h = 4



12 nodes

h = 5



20 nodes

$$h(5) = \underline{h(4)} + \underline{h(3)} + 1$$
$$= 12 + 7 + 1$$
$$= \underline{20}$$

h = 6

$$h(6) = \underset{20}{\underline{h(5)}} + \underset{12}{\underline{h(4)}} + 1$$
$$= \underline{33} \text{ nodes}$$

:- minimum # of nodes required with height (H) in AVL Tree ⇒

If H = 1, = 2
H = 2, = 4

$$\boxed{\min(H) = \min(H-1) + \min(H-2) + 1}$$