# Deletion of a node from AVL Tree

→ Deletion from the AVL tree is similar to BST

- No child
- One child
- two child.

→ Compute the balance factor

→ Consider A is the nearest ancestor of deleted node

→ if deleted node are from left subtree of A, Called Type-L deletion otherwise type R deletion.

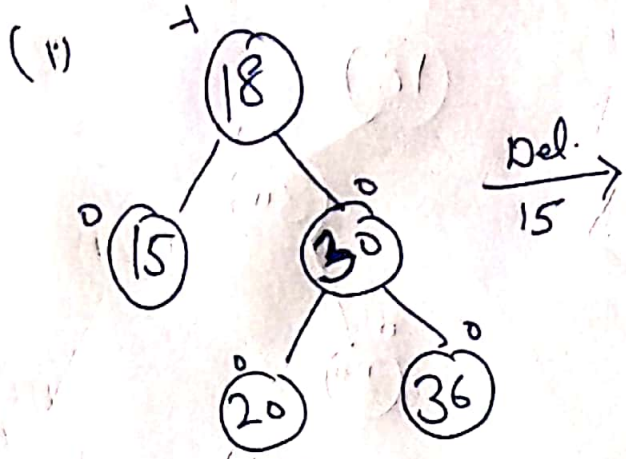
## Type L

L(0)

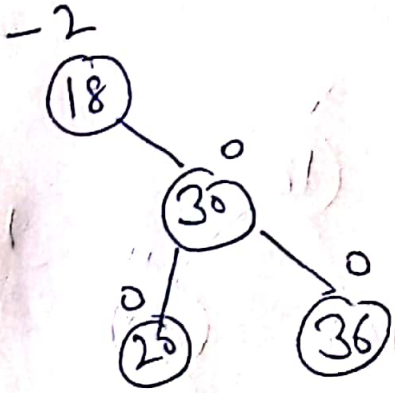
L(1)

L(-1)

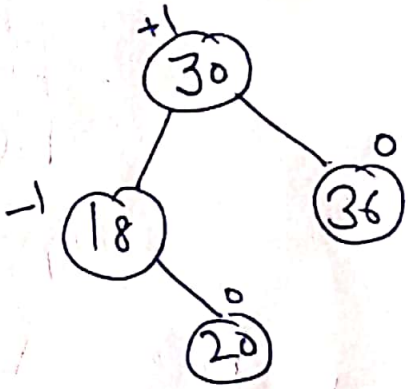
(i)



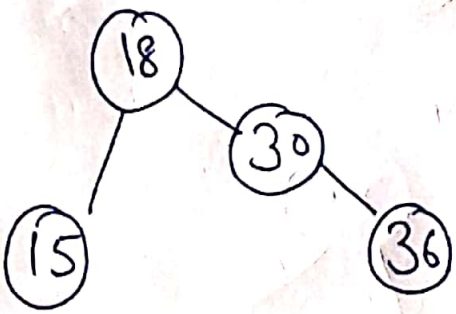

Del. 15 →



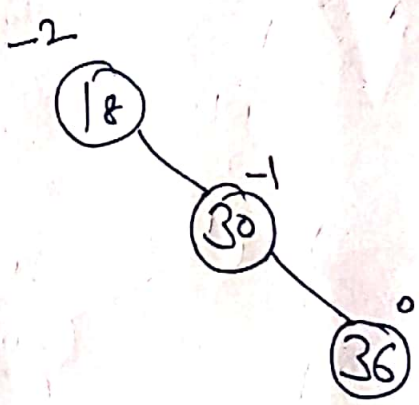

L(0) Perform Left rotation ↓



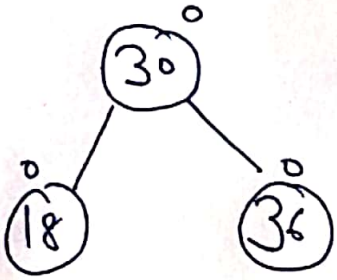

(ii)

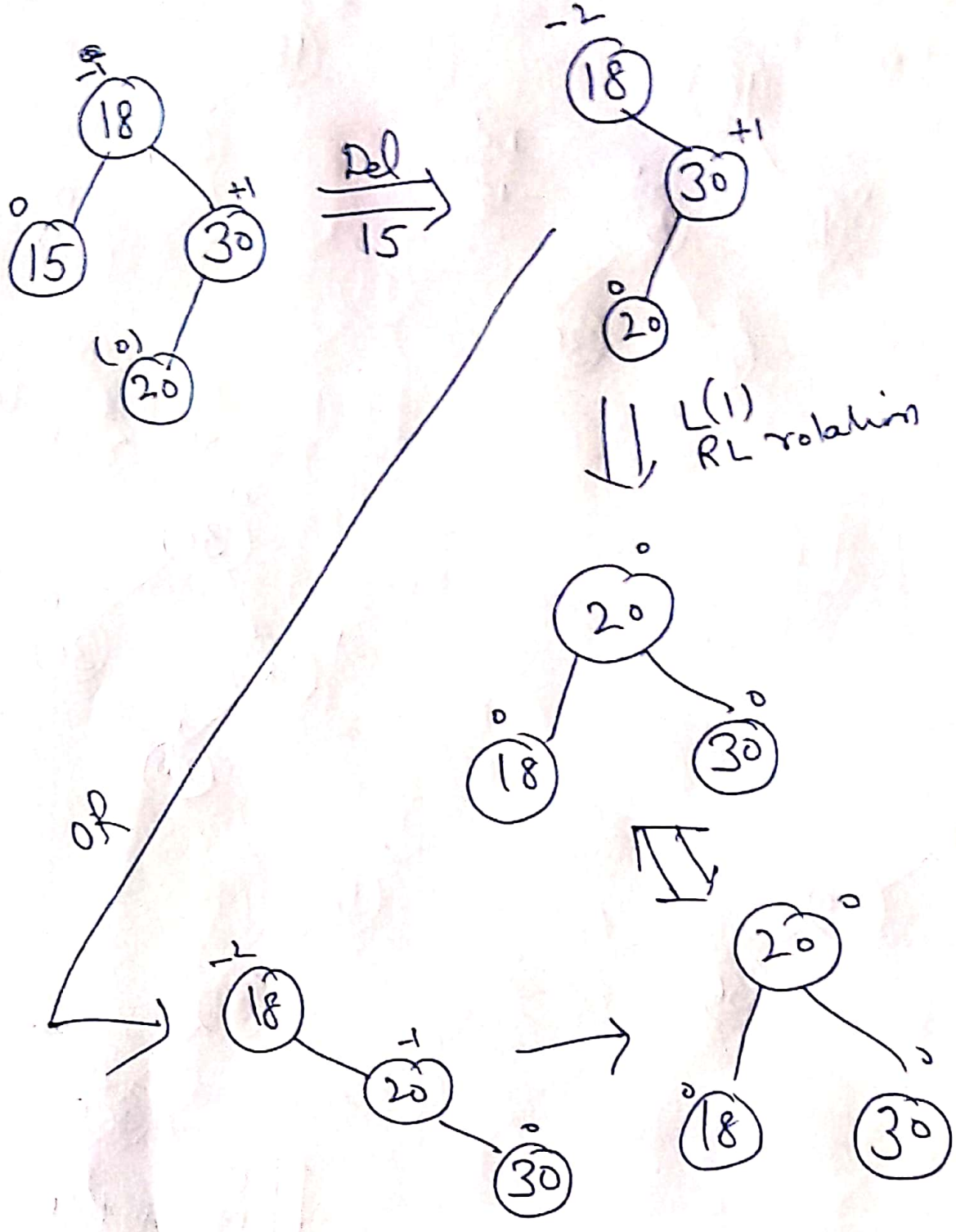



Del 15 ⇒



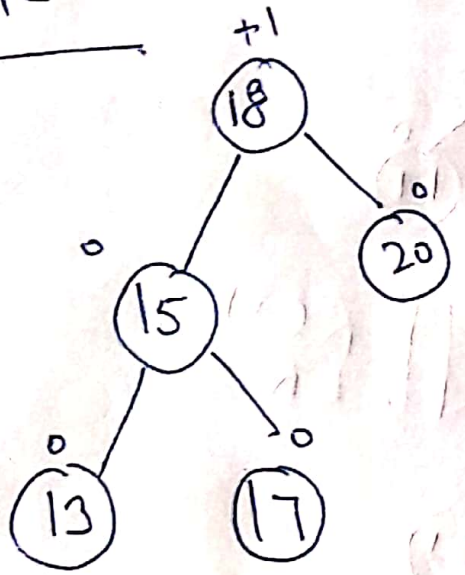

L(-1) Perform Right rotation ↓



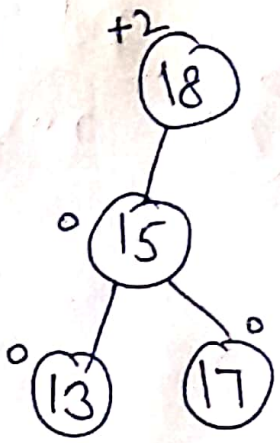


Type R

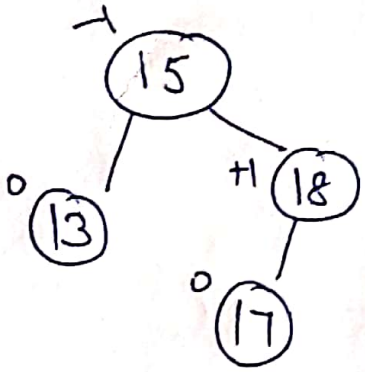
(i)




Del. 20 →

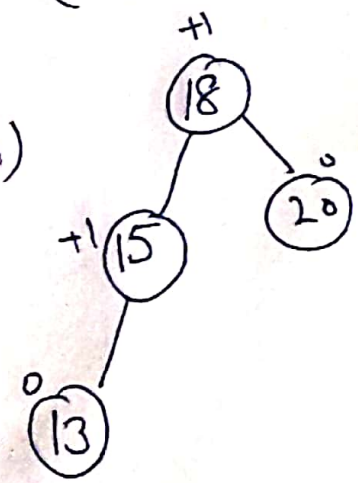



R(0)
Right rotation over node A

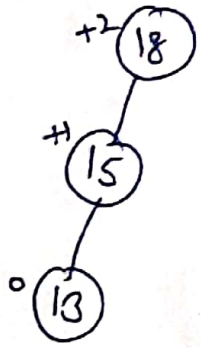



Type R
(i) R(0)
(ii) R(1)
(iii) R(-1)

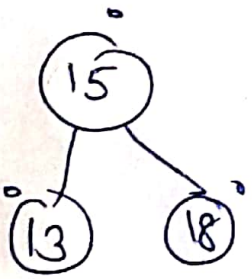
(ii)



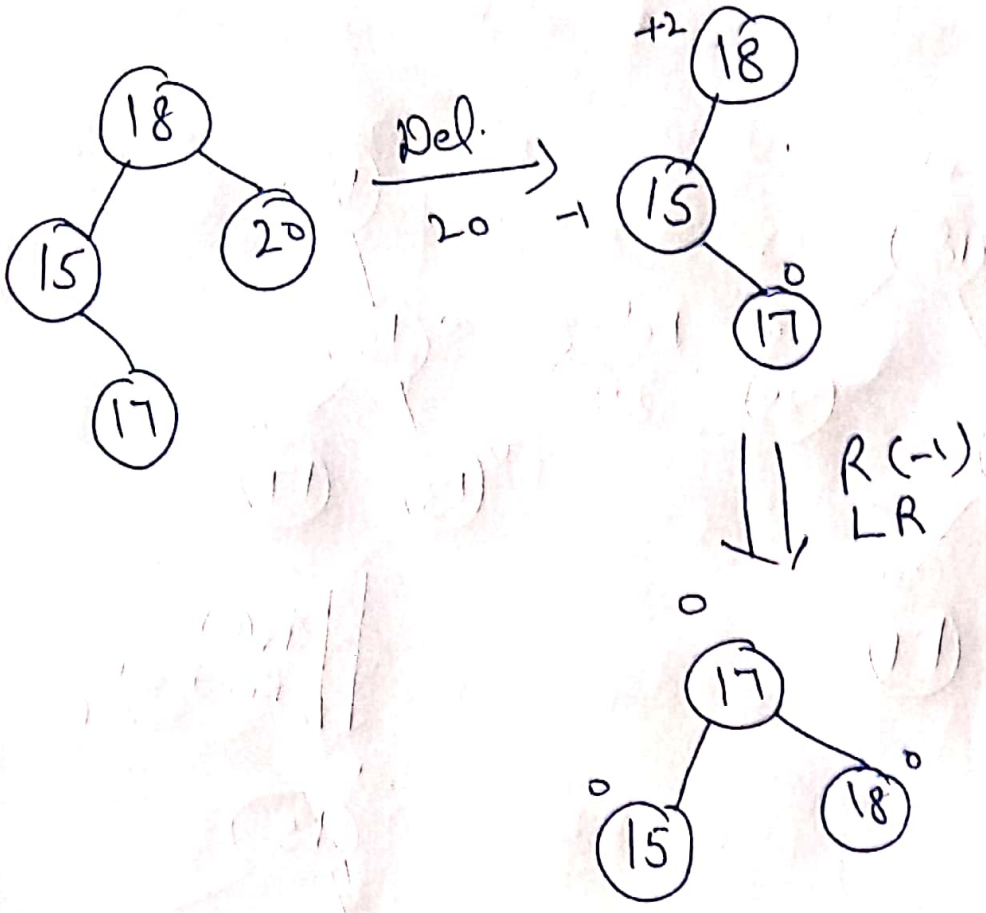

Del. 20 →

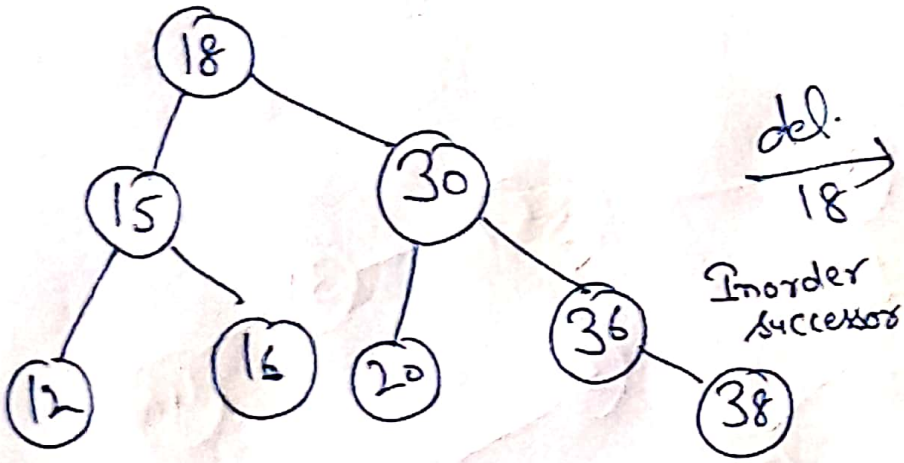



R(1)
Right rotation.



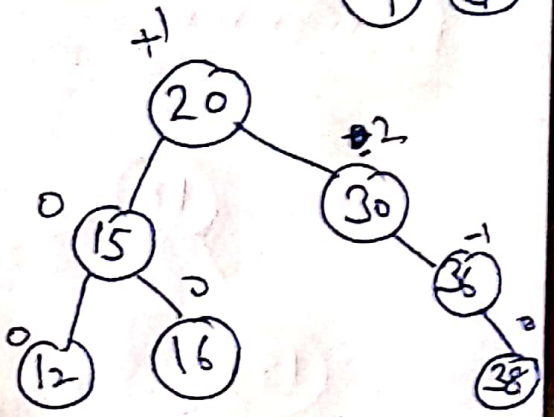
(iii)






del. 18

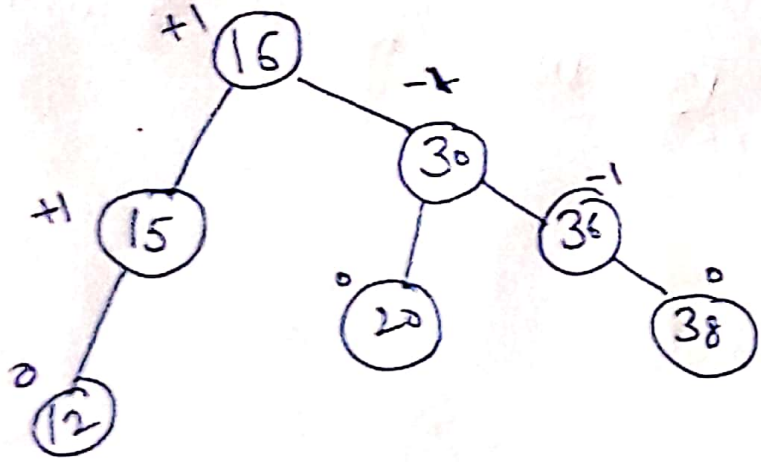
Inorder successor

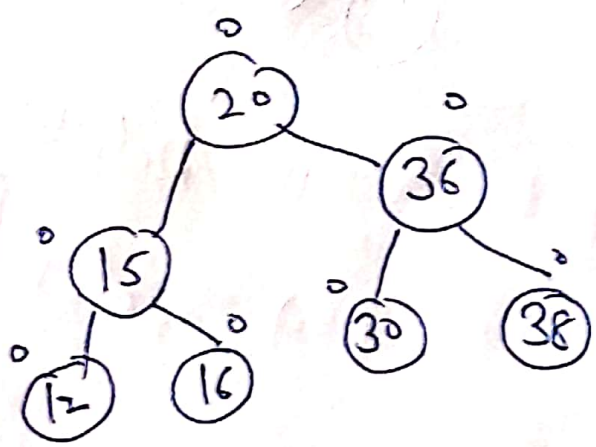



L(-1)

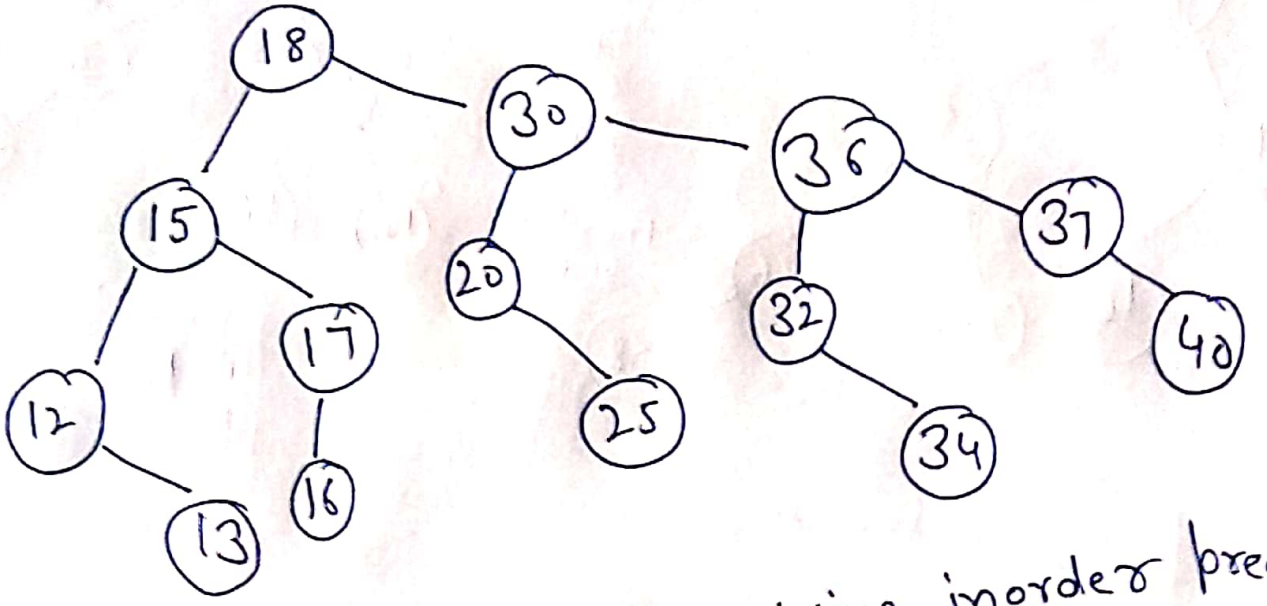
Perform left rotation about 30

del(18)
Inorder Predecessor



Assignment :




Perform (i) Delete 15 using inorder predecessor

(ii) Delete 13 using inorder Successor.