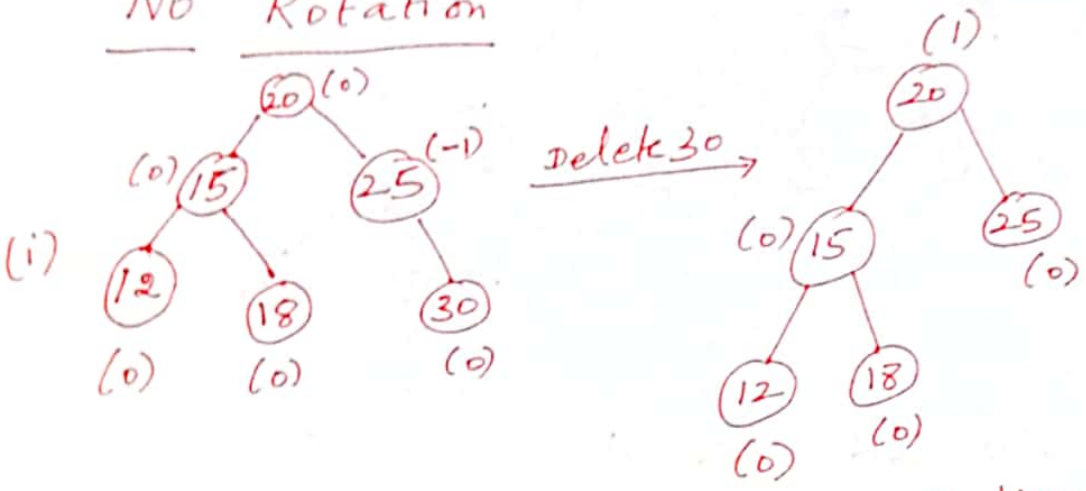
AVL TREE

DELETION

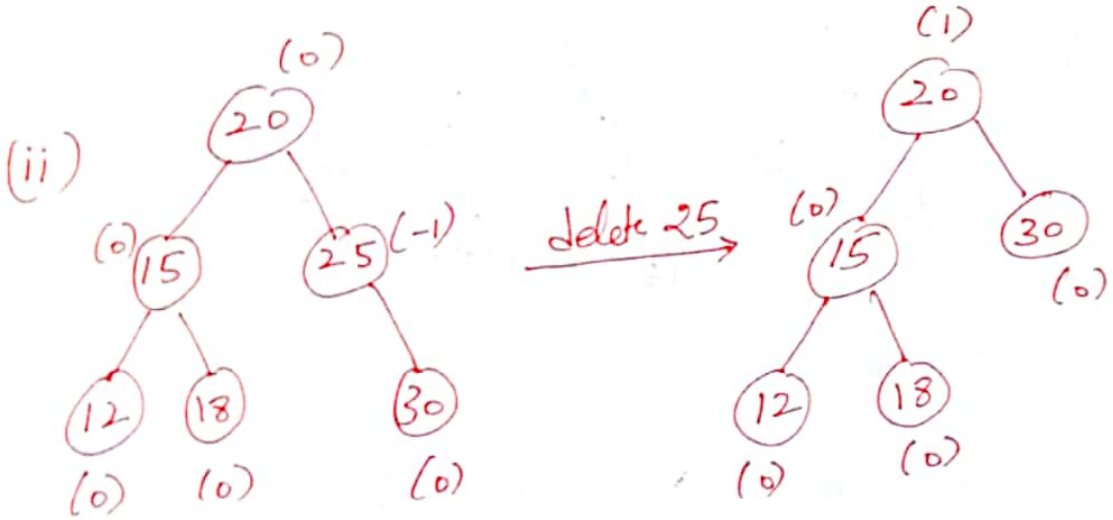
No Rotation

(i)



Balanced. Hence No rotation.

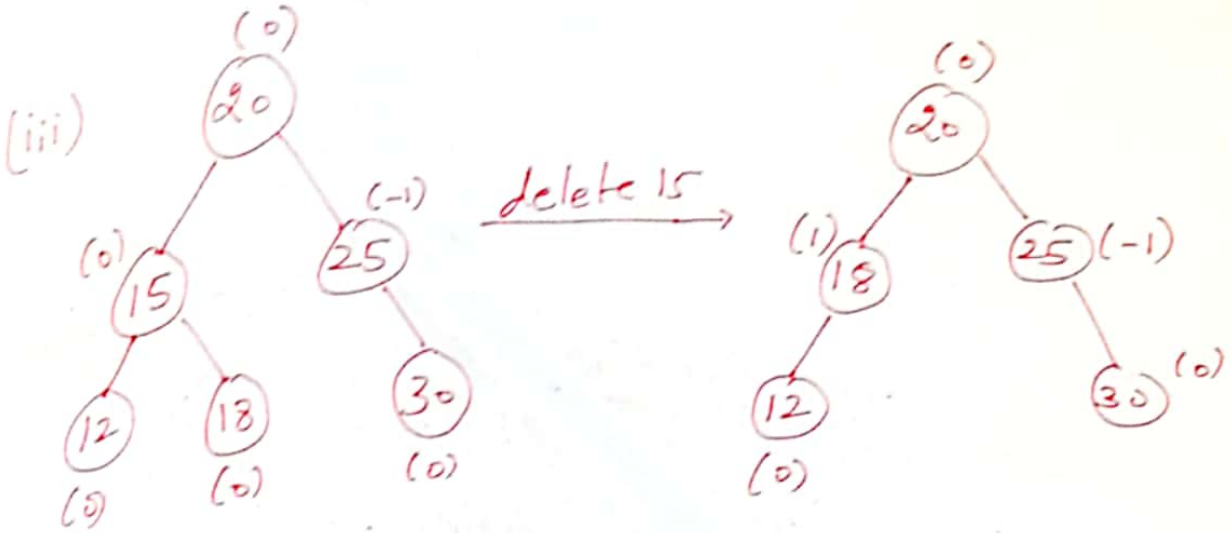
(ii)



Balanced. Hence No Rotation

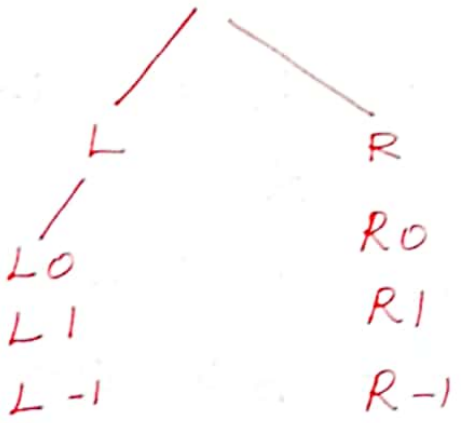
Scanned with CamScanner

(iii)



Balanced. No Rotation

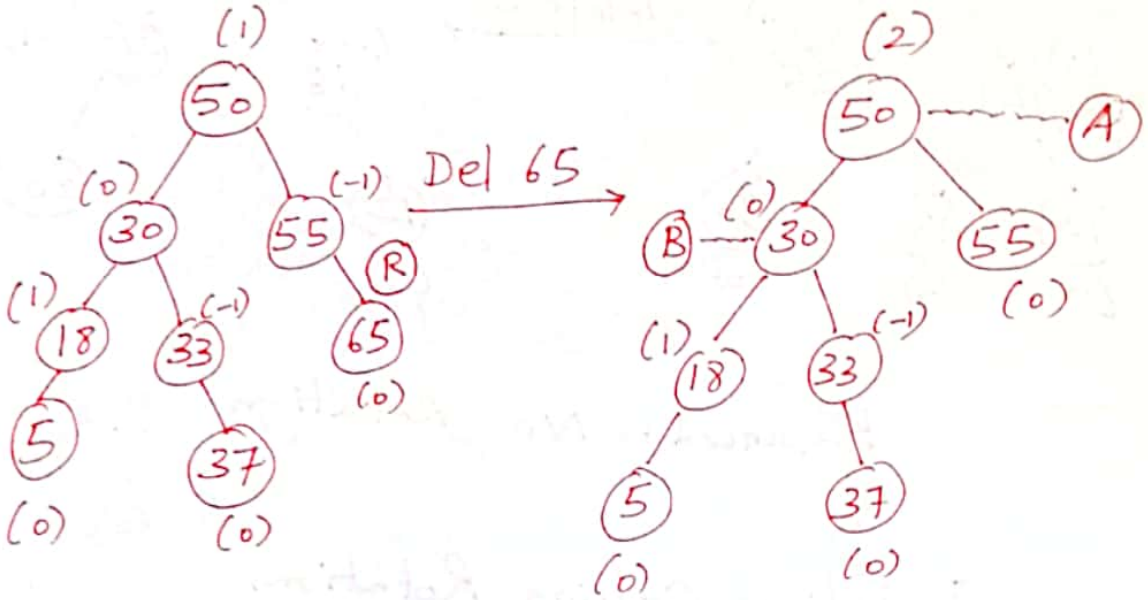
Deletion Calling Rotations



R0 Rotation

Scanned with CamScanner

Example



Deletion is Right Child of Node 'A'.

Hence 'R'.

Left subtree root of A is called 'B'.

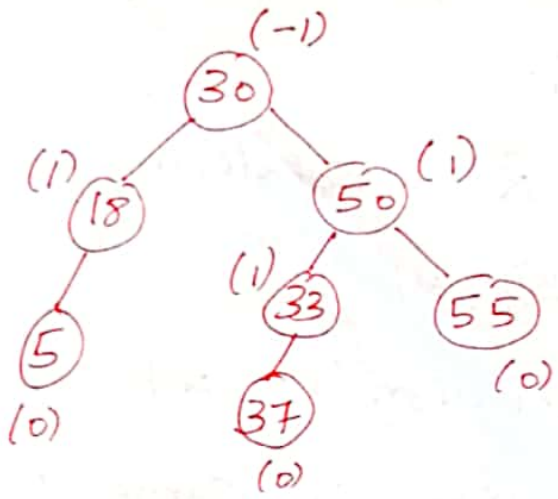
∴ Bf of B is 0.

Hence classification is R0.

R0 Rotation is LL over A w.r.t. B.

Scanned with CamScanner

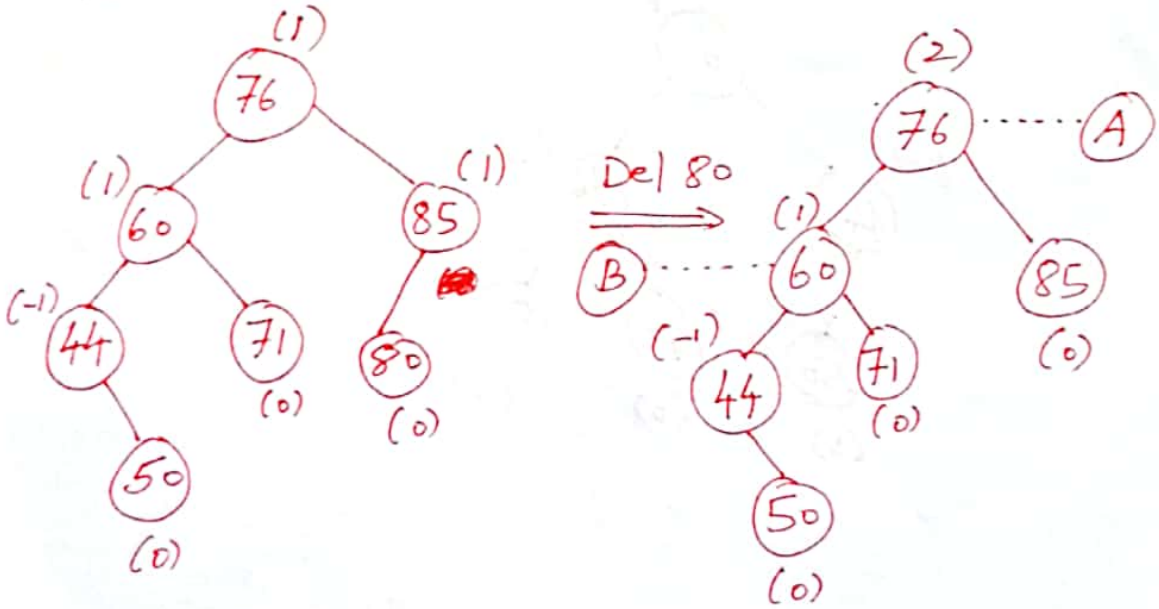
Doing RO



Balanced.

R1 Rotation

Example



Scanned with CamScanner

Deletion is Right Child of Node A

∴ R Rotation

Bf (B) = 1

∴ Rotation is R1

R1 Rotation is same as R0

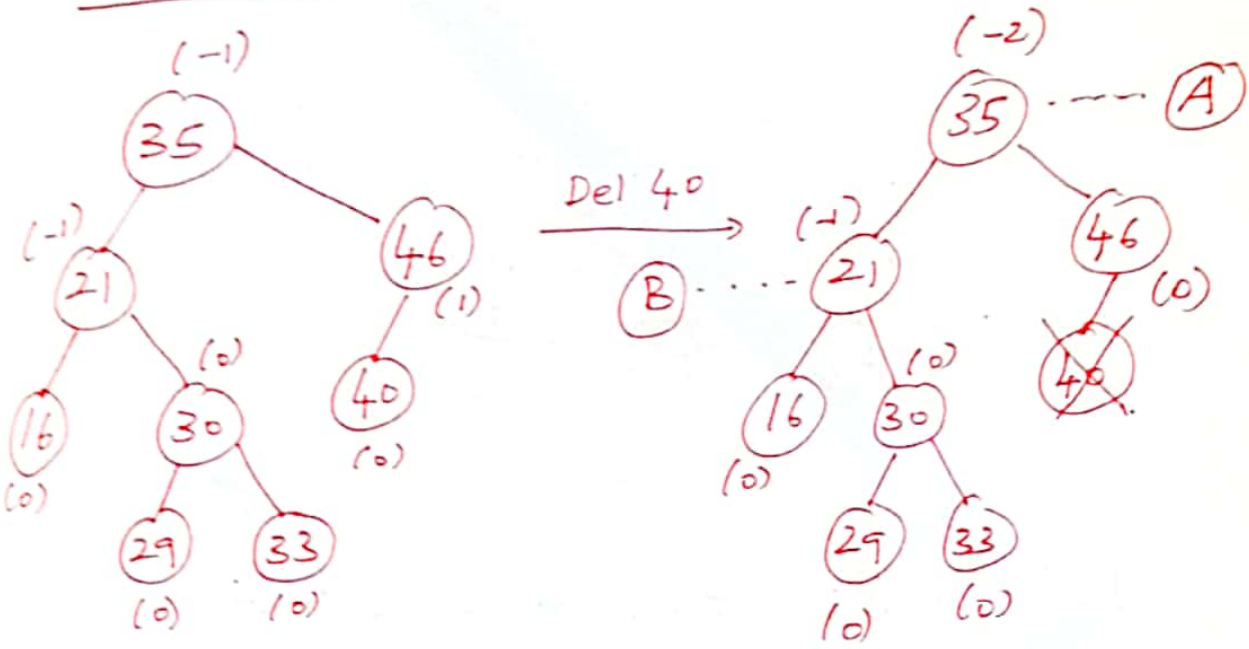
(ie) R1 is LL over A w.r.t. B.

Doing R1

Scanned with CamScanner

R-1 Rotation

Example



Deletion is Right child of A.

∴ R Rotation.

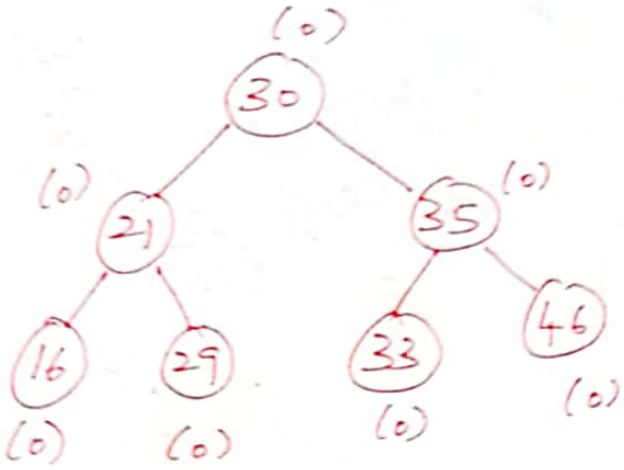
Bf (B) = -1

∴ R-1 Imbalance

Doing LR over A.

Scanned with CamScanner

Doing R-1 Rotation



∴ R0, R1 - LL over A

R-1 - LR over A

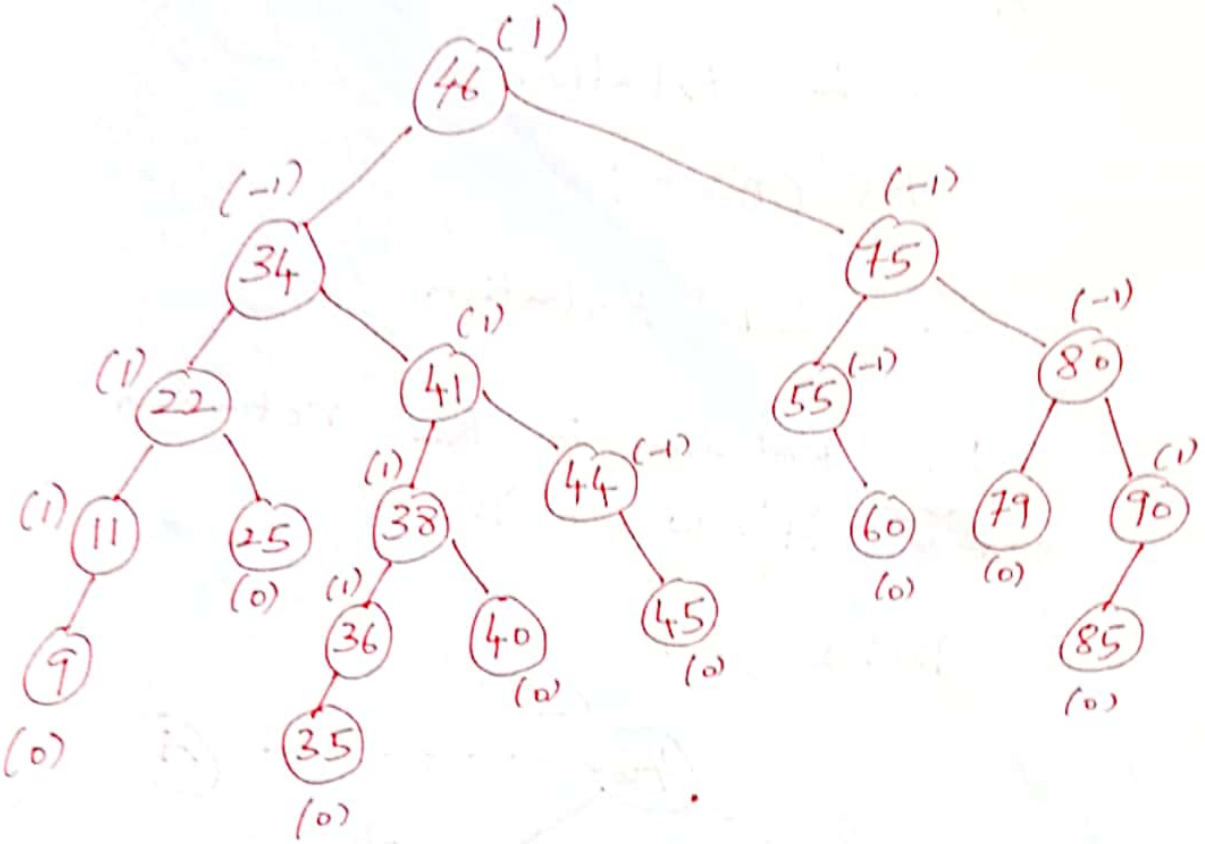
Similarly,

L0, L-1 - RR over A

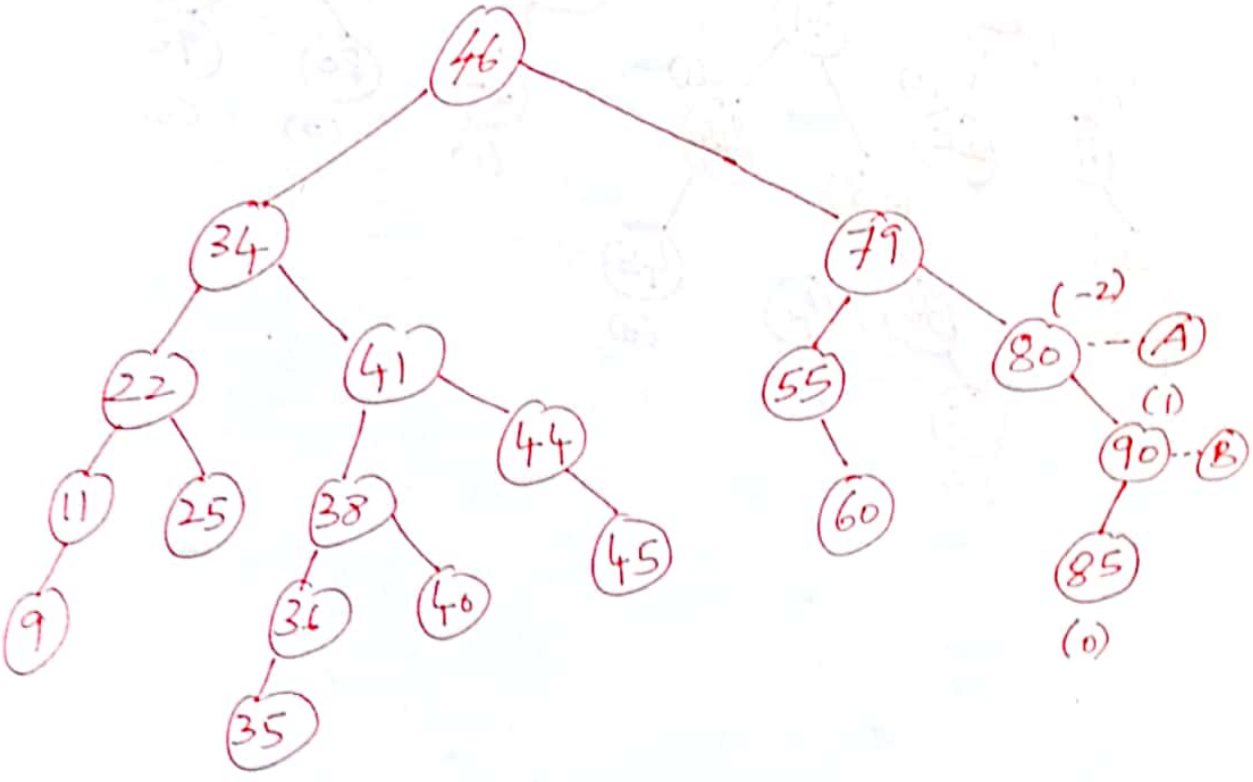
L1 - RL over A

Scanned with CamScanner

Delete 75.



Deleting 75



Scanned with CamScanner

Deletion is done to Left of A.

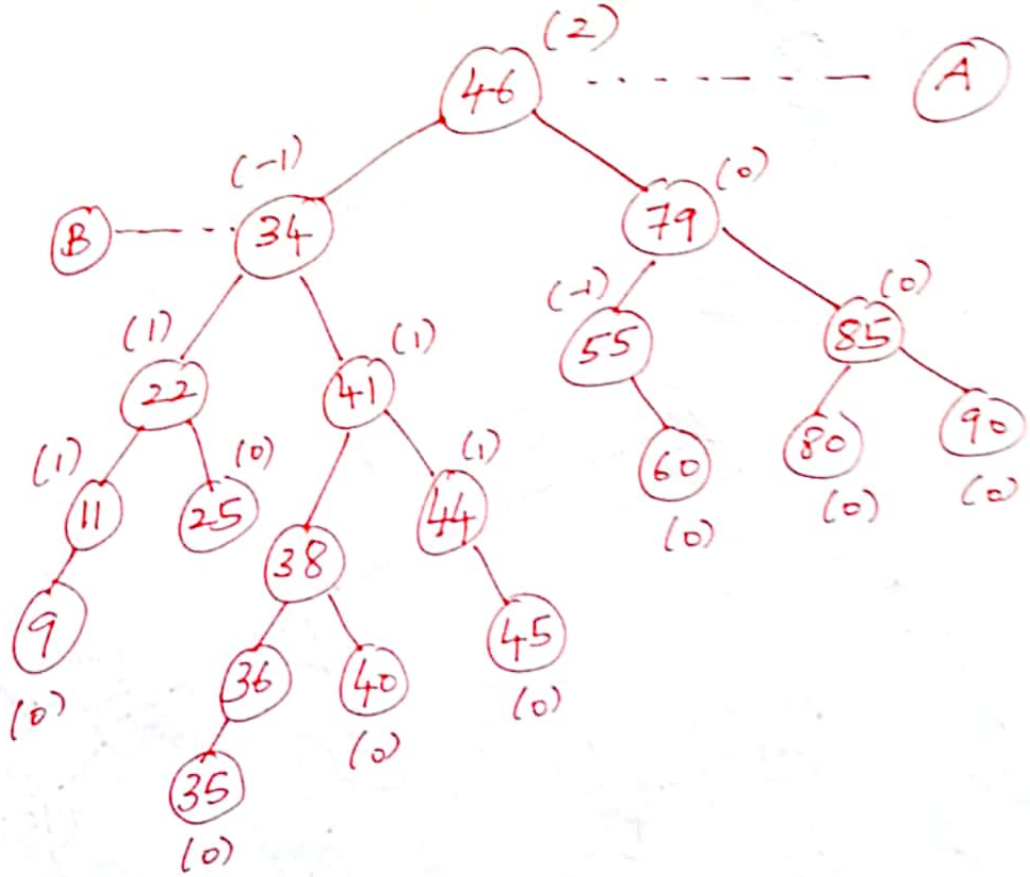
∴ L Rotation.

Bf (B) = 1

∴ L1 Rotation.

L1 Rotation is RL rotation over 'A'. w.r.t. 'B'.

Doing L1



Scanned with CamScanner

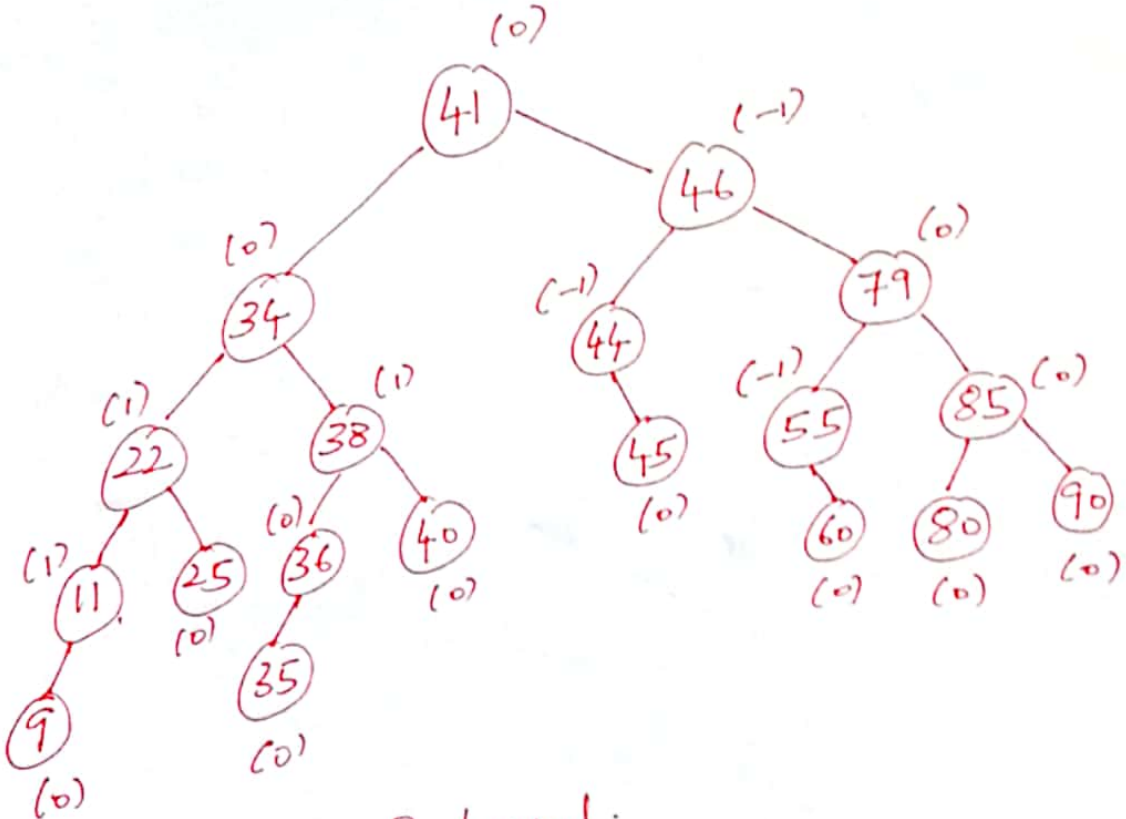
Deletion is done to Right of A.

∴ R Rotation

Bf (B) = -1

∴ R -1 Rotation.

R -1 Rotation is LR over A w.r.t. B.



Balanced.

Scanned with CamScanner