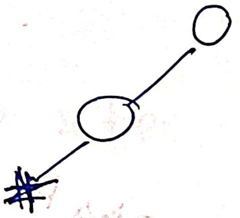
Main intuition behind AVL tree.

insertion → if (balanceFactor > 1 && $\underline{\text{node.left.key > key}}$)

⇒ LL case

this instruction tells us where the key got deleted.

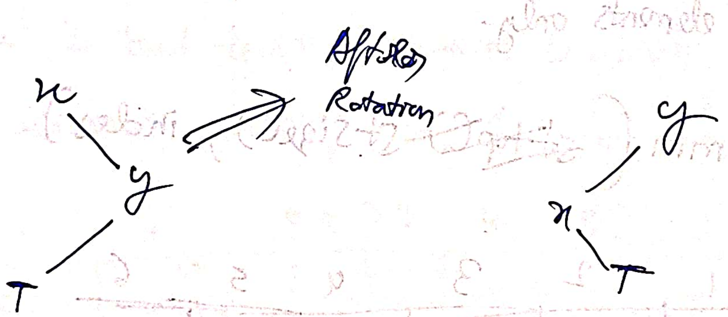


deletion → if (balanceFactor > 1 ~~&&~~ && $\underline{\text{balanceFactor(node.left)} \geq 0}$){

} LL case

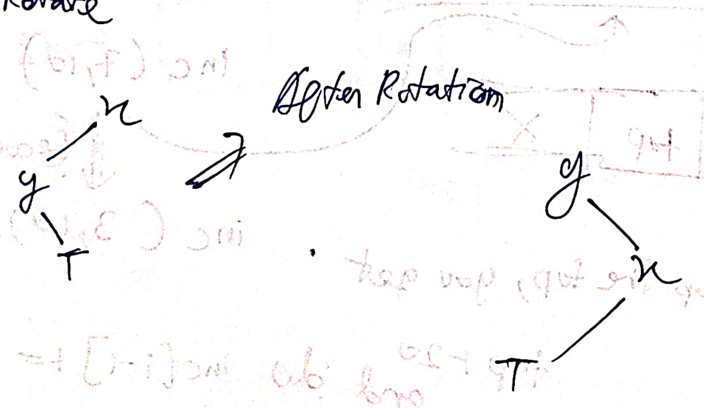
No way to check where deletion happened so we do generic checking

leftRotate



(Since x is to become y's left child hence keeping a reference of y's current left child is imp)

rightRotate



(Since y's right child is going to be x, so we need a pointer to y's right child already).