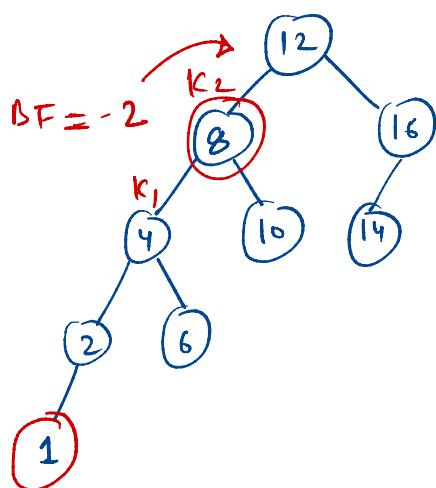
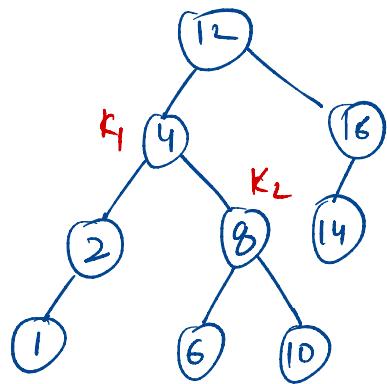


AVL Trees - III

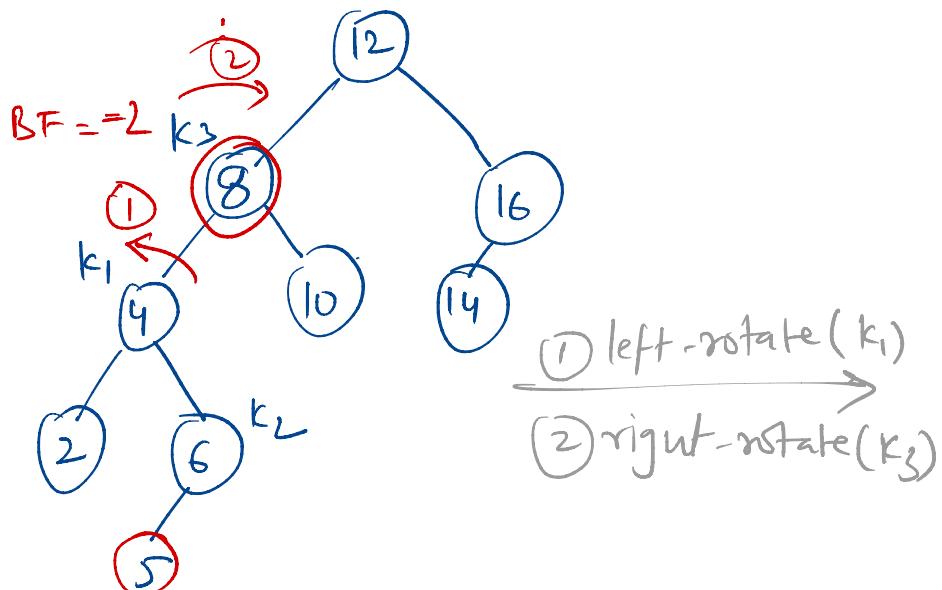
Examples : Case LL (right rotation)



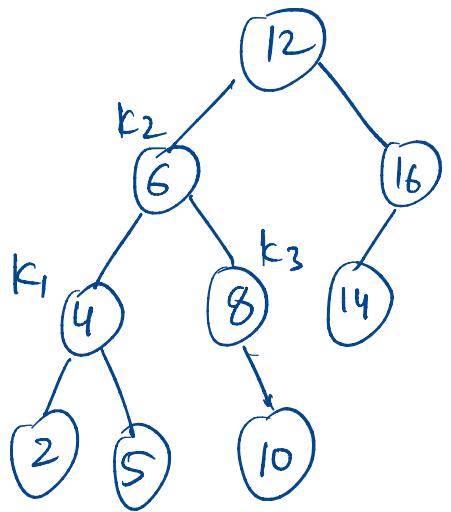
right-rotate(k_2)



Case LR (left-right double rotation)



(1) left-rotate(k_1)
(2) right-rotate(k_3)



Implementing Add in an AVL Tree

- Perform normal BST add. When the recursive call returns, balance the node if there is an imbalance
→ i.e., if BF becomes $-2/+2$.

- logic to detect the case for resolution:

- Current node $\underline{BF < -1} \Rightarrow$ Case LL or LR

→ look at current node's left child BF

fix with R rotation | left child $BF < 0 \Rightarrow$ Case LL

fix with LR rotation | left child $BF > 0 \Rightarrow$ Case LR

- Current node $BF > 1 \Rightarrow$ Case RL or RR

→ look at current node's right child BF

fix with RL rotation | Right child $BF < 0 \Rightarrow$ Case RL

fix with R rotation | Right child $BF > 0 \Rightarrow$ Case RR

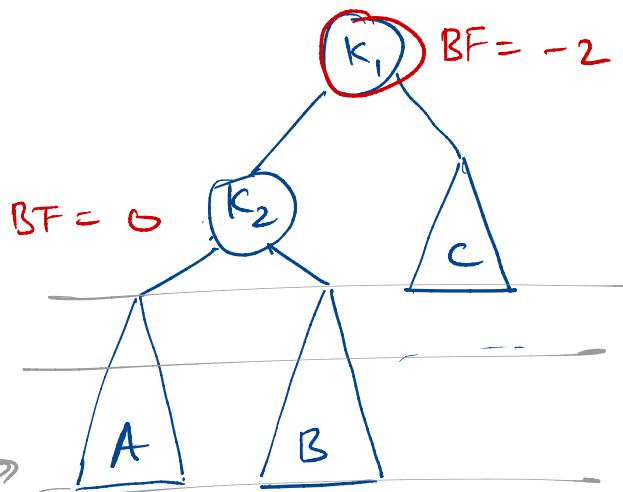
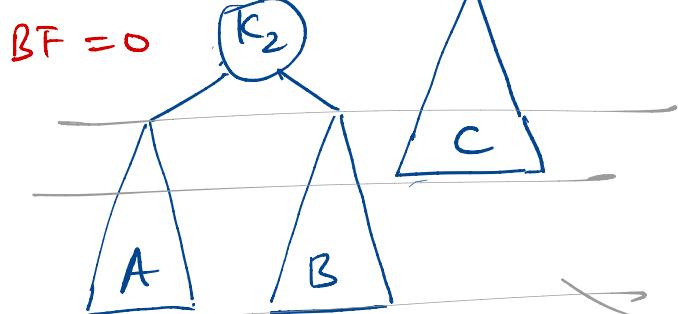
AVL remove Cases:

→ Similar cases as with adding : LL, LR, RL, & RR

→ Two additional case beyond these

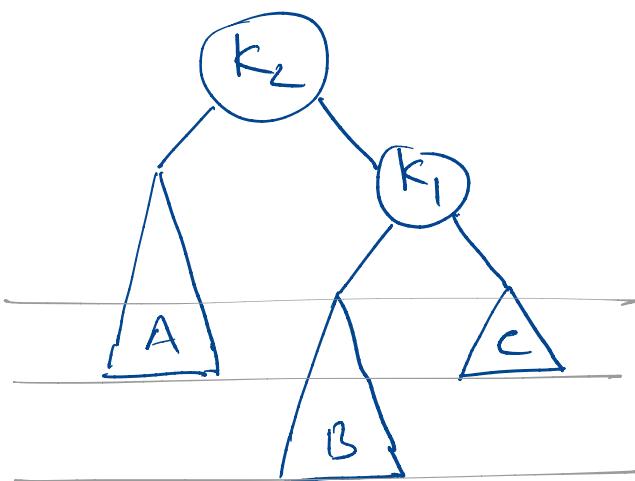
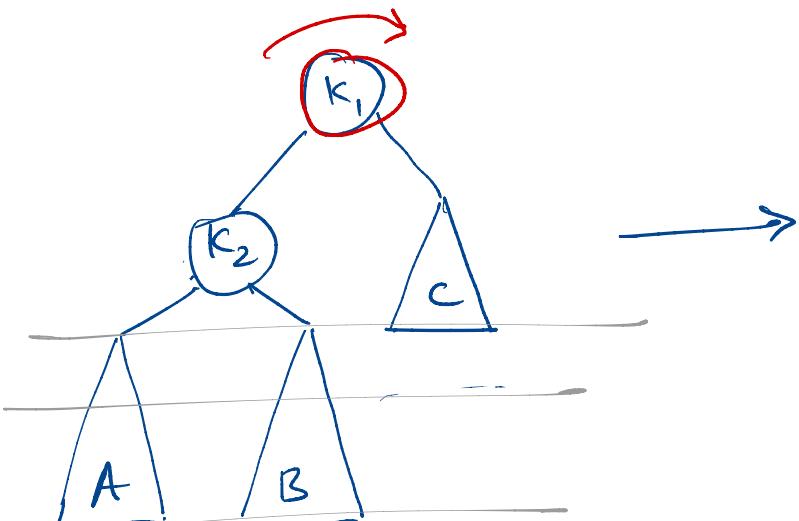
→ Offending subtree has a balance factor of 0

Case 5 : L-both k_1 $BF = -1$
 $H(A) = H(B) = H(c)$

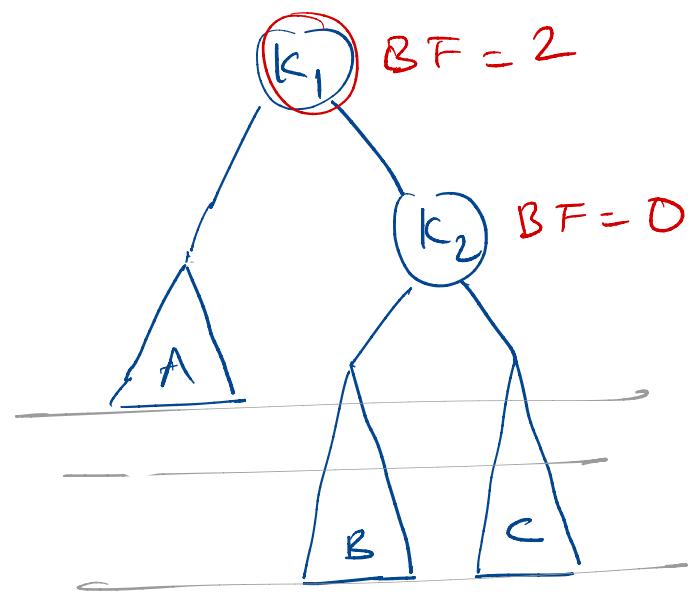
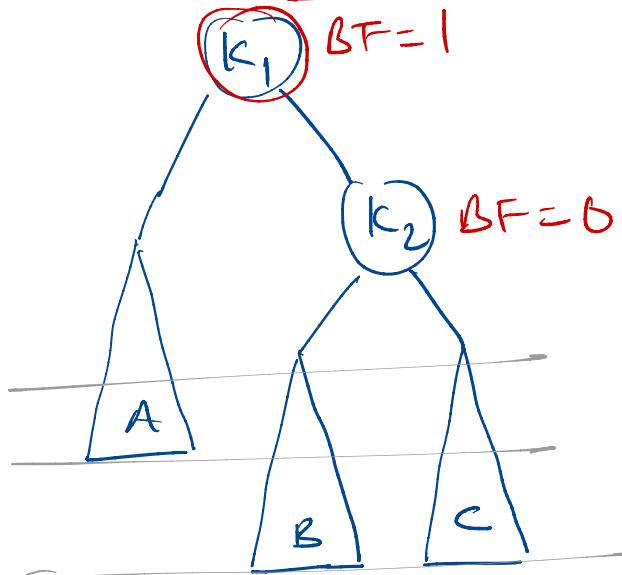


remove from C

Resolution: Perform a right rotation about k_1 .

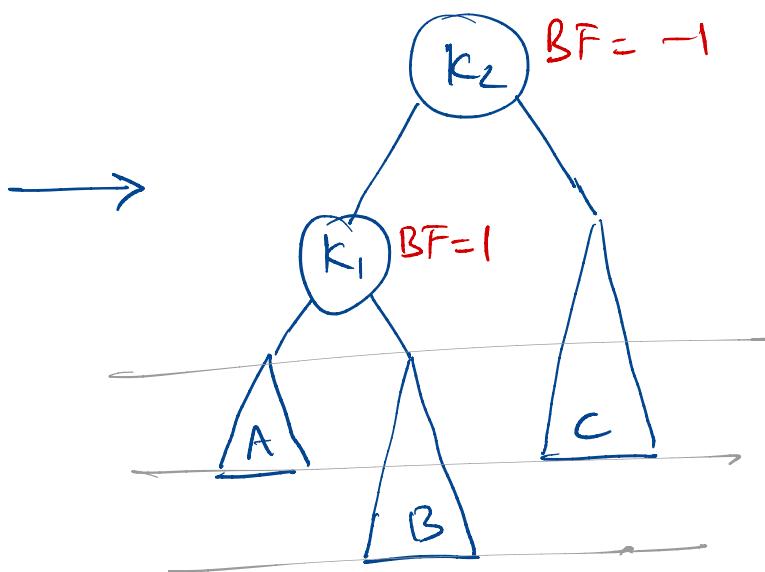
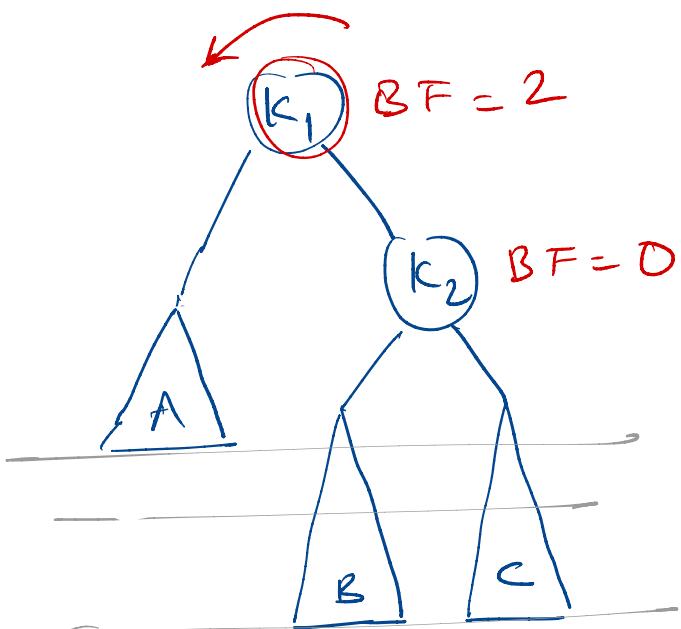


Cone '6' \circ R-both



Remove from A

Resolution: Rotate left about k_1 ,



Implementing remove in a AVL tree :

→ perform normal BST remove and fix imbalance at a node while going up

→ Logic to detect the case for resolution:

- Current node $BF < -1 \Rightarrow$ Case LL, LR, or L-both
 - look at current node's left child BF

fix with R rotation

left child $BF < 0 \Rightarrow$ case LL

fix with LR rotation

left child $BF > 0 \Rightarrow$ case LR

fix with R rotation

left child $BF = 0 \Rightarrow$ case L-both

- Current node $BF > 1 \Rightarrow$ Case RR, RL or R-both

→ look at current node's right child BF

fix with RL rotation

Right child $BF < 0 \Rightarrow$ case RL

fix with L rotation

Right child $BF > 0 \Rightarrow$ case RR

fix with L rotation

Right child $BF = 0 \Rightarrow$ case R-both

Complexity : Adding $O(\lg n)$

removing $O(\lg n)$