

# AVL Trees, AVL Sort

## □ BST Recall

$h$ : height of the tree;  
longest path from root to leaf.

### ◦ Balanced or NOT

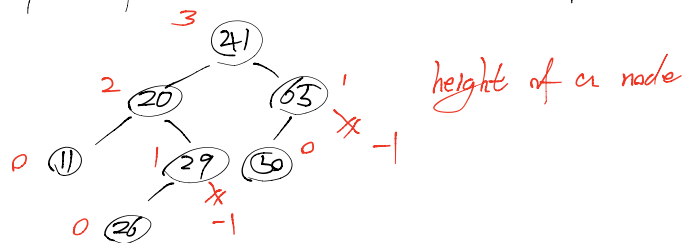
• Balanced: if  $h = \Theta(\lg n)$

• keep tree balanced

### ◦ Height of node in a tree

longest path from it down to a leaf.

e.g.



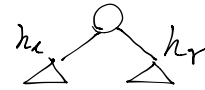
$$\begin{aligned} \text{height of node} &= \text{height of left child} \\ &+ \\ &\text{height of right child} \\ &+ \\ &1 \end{aligned}$$

## AVL Tree

Adelso - Velski - Landi's

• AVL Tree require heights of left & right children of every node to differ by most  $\pm 1$

$$|h_l - h_r| \leq 1$$

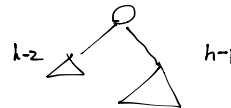


• AVL trees are always balanced ( $\Theta(\lg n)$  tree height)

worst case is when right subtree has height one more than the left for every node.

Define:  $N_h \equiv$  min #nodes in an AVL tree of height  $h$

$$N_h = 1 + N_{h-1} + N_{h-2}$$



$$N_h > F_n(\text{Fibonacci}) = \frac{\varphi^h}{\sqrt{5}} \quad (\varphi = 1.618)$$

$$\hookrightarrow \frac{\varphi^h}{\sqrt{5}} < N_h$$

$$\hookrightarrow h < \log_{\varphi} n \approx 1.44 \lg n$$

$$N_h = 1 + N_{h-1} + N_{h-2} > 1 + 2N_{h-2}$$

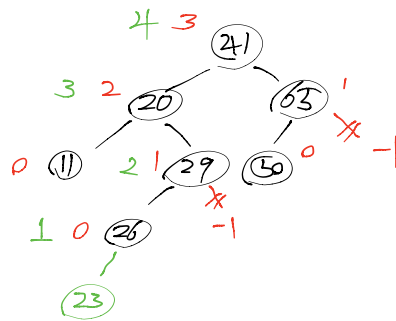
$$> 2N_{h-2} = \Theta(2^{h/2})$$

$$\hookrightarrow h < 2 \lg n$$

## AVL Insert

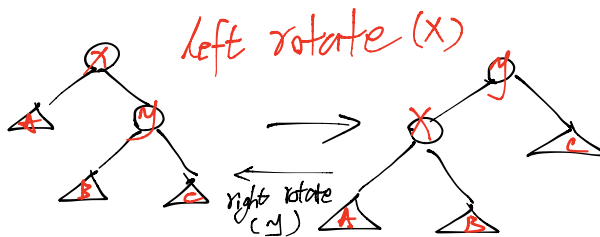
- ① Simple BST insert
- ② fix AVL property

e.g.



Insert (23)

o Rotation:



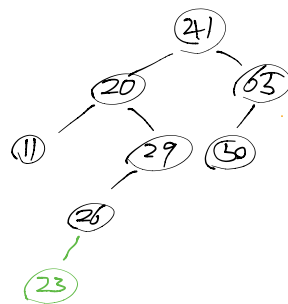
Const. time

(root moves to the left)  
↳ left rotate

AXBYC → AXBYC

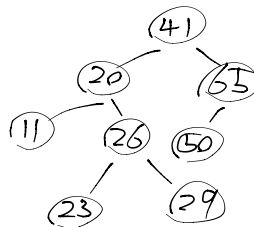
o AVL Insert (Continue)

e.g.



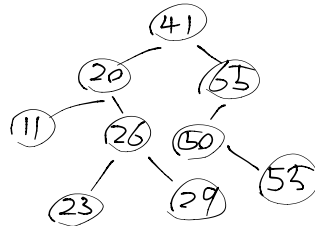
Insert (23)

right rotate (29)

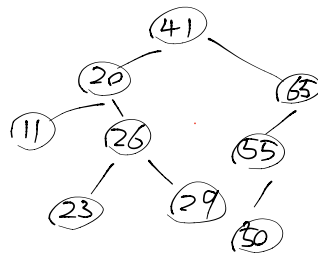


AVL tree !

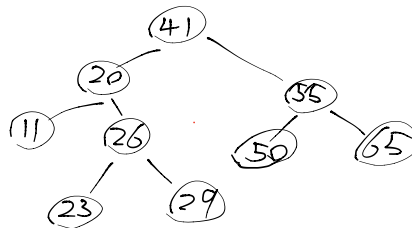
e.g: Insert (55)



left  
rotate (50)



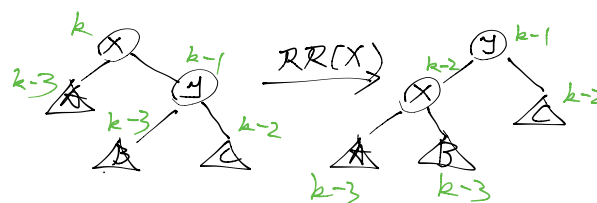
right rotate (65)



② fix AVL Property

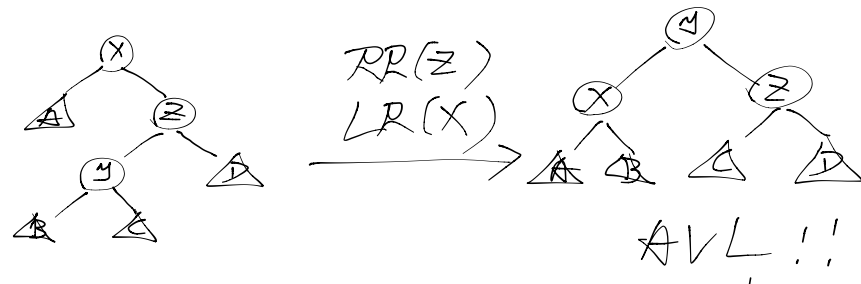
from changed node up, suppose  $x$  is lowest node violating AVL, assume  $x.right$  is heavier

— if  $x$ 's right child is right-heavy or balanced  
do Right rotate of  $x$ :  $RR(x)$



balanced

— else do double rotation



## □ AVL Sort

- Insert  $n$  items  $\Theta(n \lg n)$
- In-order traversal  $\Theta(n)$

AVL is another powerful tool to do sort in  $\Theta(n \lg n)$  time

## ○ Abstract Data Type

- Insert / delete } priority queue
- min
- successor / pred

