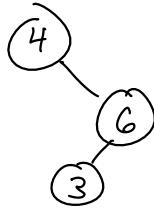


What is a binary tree?

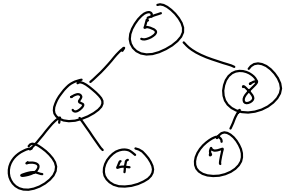
A tree where each node has at most 1 left child and 1 right child.

What is a BST?

A binary tree where ^{every} left descendant is smaller and ^{every} right descendant is greater for all nodes.

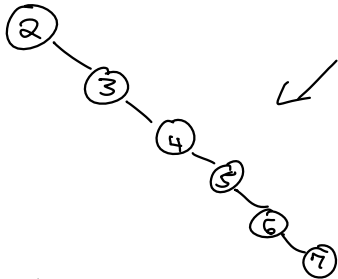


NOT A
BST



This is
a BST

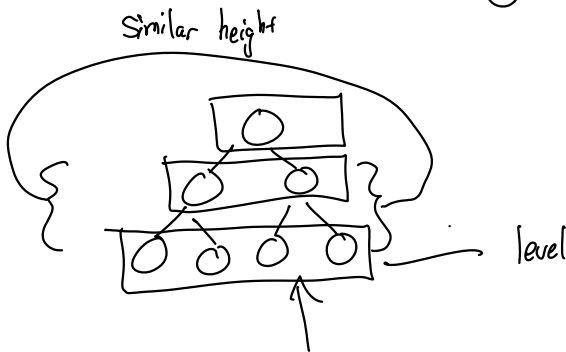
get — $O(\log n)$ for nice trees



This is
also a
BST.

get — $O(n)$ for bad trees.

||
∩



get — $O(\text{height})$ for any BST

Goal: height to be $O(\log n)$

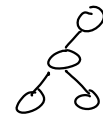
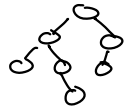
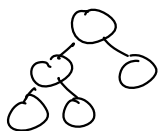
Good tree b/c height is small in comparison to # of nodes
 $O(\log n)$

A tree is full if every level has max # of nodes or zero nodes.

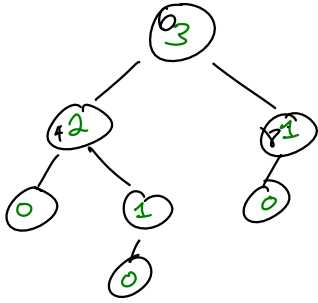
A full tree has $2^{h+1} - 1$ nodes.

AVL tree:

A BST s.t. for every node, height of left subtree and height of right subtree differ by at most one.

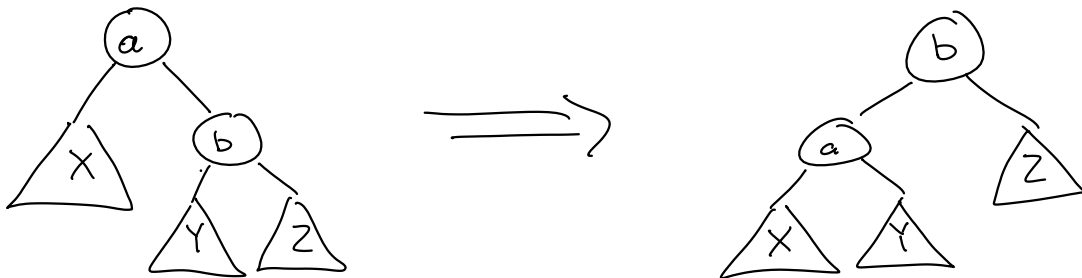


We will track height at every node.



Tree Rotations

Left rotation:



Function rotateLeft(tree):

$a \leftarrow \text{tree}$

$b \leftarrow \text{tree} \rightarrow \text{right}$

$X \leftarrow \text{tree} \rightarrow \text{left}$

$Y \leftarrow b \rightarrow \text{left}$

$Z \leftarrow b \rightarrow \text{right}$

$a \rightarrow \text{right} \leftarrow Y$

$b \rightarrow \text{left} \leftarrow a$

Return b

EndFunction

$O(1)$

Prove: rotateLeft(tree) produces a BST if tree is a BST

- $X < a$
- $Y < b$
- $Z > b$
- $a < b$
- $a < Y$

Method insertInSubtree (key, value, node) :

If node is an empty tree:

Return new Node (key, value)

Else :

If node \rightarrow key < key:

node \rightarrow right \leftarrow insertInSubtree (key, value, node \rightarrow right)

node \rightarrow height \leftarrow max (height of node \rightarrow left, height of node \rightarrow right) + 1

node \leftarrow rebalance (node)

Else If node \rightarrow key > key:

node \rightarrow left \leftarrow insertInSubtree (key, value, node \rightarrow left)

node \rightarrow height \leftarrow max (height of node \rightarrow left, height of node \rightarrow right) + 1

node \leftarrow rebalance (node)

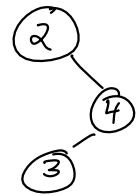
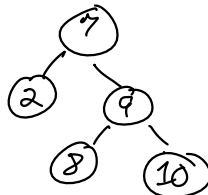
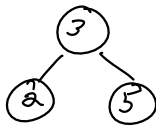
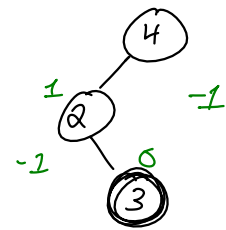
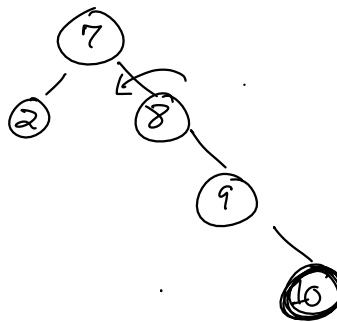
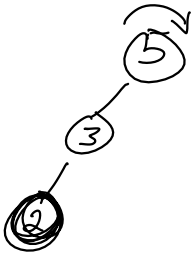
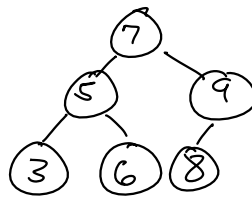
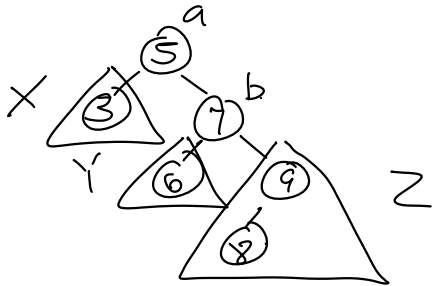
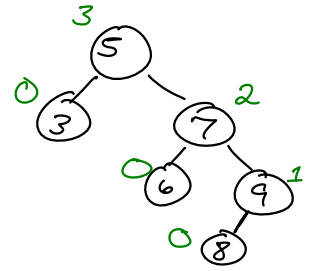
Else :

||

End If

End If

End Method



||