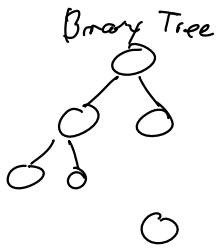
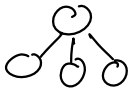
What is a binary tree?

A tree where each node has at most one left child, and at most one right child



NOT BT



What is a BST?

A binary tree where, for each node, left descendents are lesser, right descendents are greater

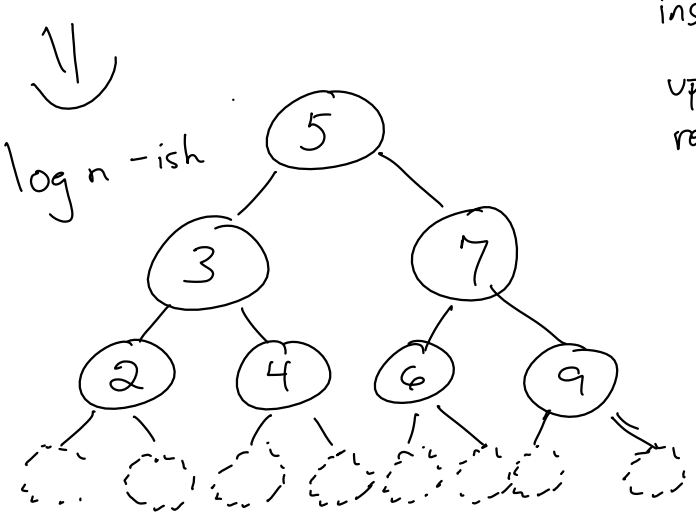
What is a BST for?

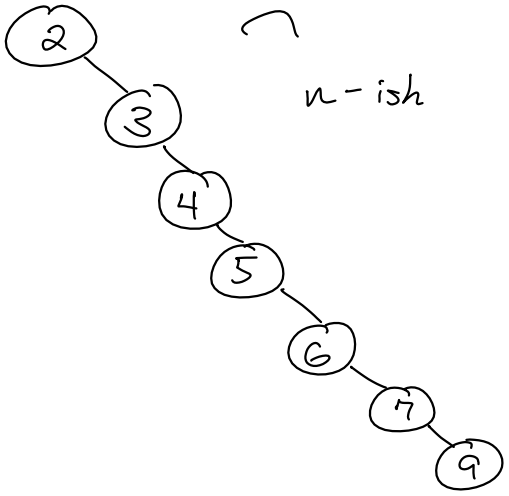
Kind of dictionary: get(K) —— O(height)

insert(K, V)

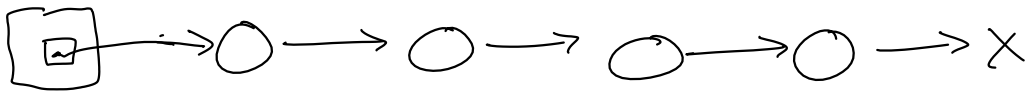
update(K, V)

remove(K)





Linked List size



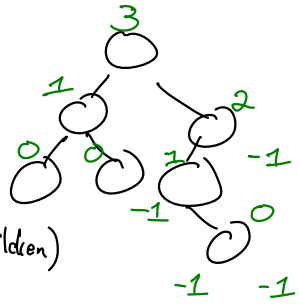
getSize w/o size field — O(n)

getSize with size field — O(1)

getHeight()

height of empty tree = -1

height of non-empty tree = 1 + max(heights of children)



to make getHeight O(1) for our purposes, height field on every node of tree

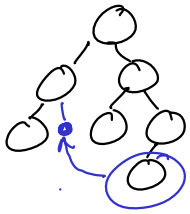
LinkedBalancedBSTNode

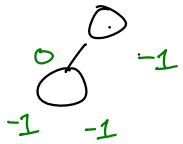
left key height
right value

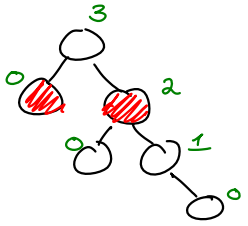
AVL Tree

AVL Tree is a BST such that, for every node, height of left and right subtrees differs by at most one.

NOT an AVL Tree

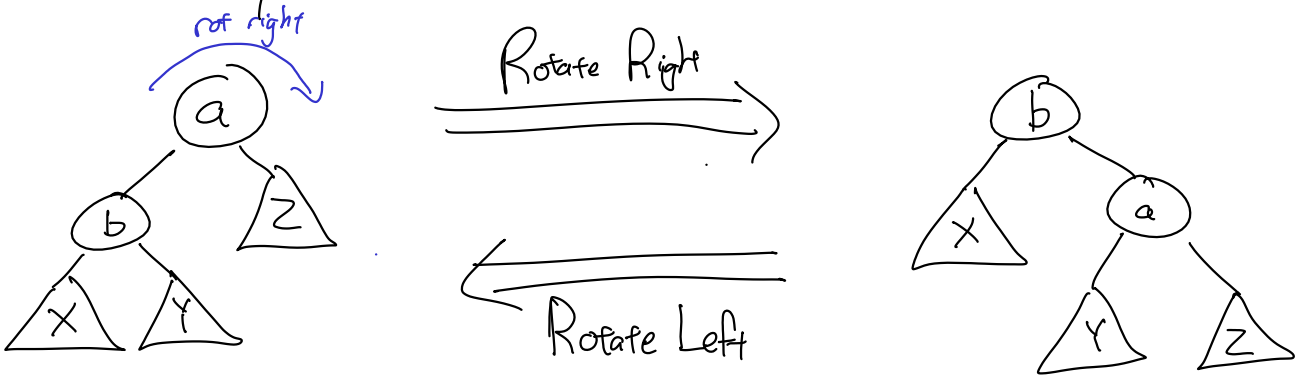






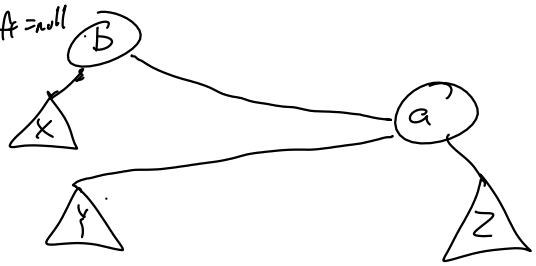
Tree Rotations

Allow us to reshape a BST in constant time

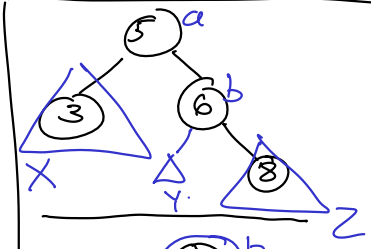


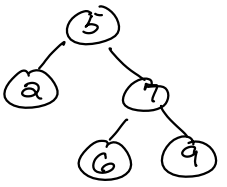

```
Function rotateRight(node):
    a ← node
    b ← node.left
    Y ← node.left.right
    a.left ← Y
    b.right ← a
    Return b
End Function
```

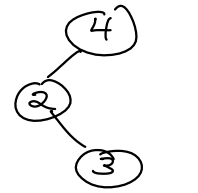
if node.left = null

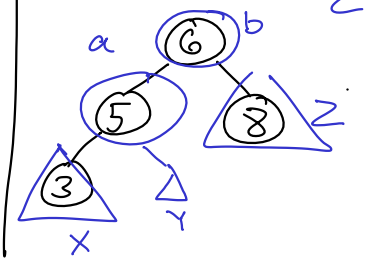


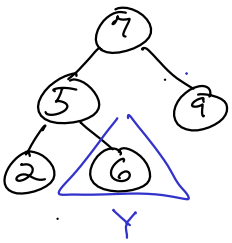
Rotate Left

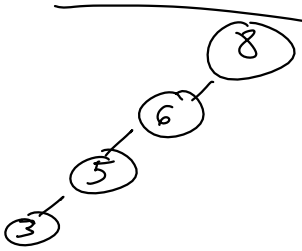


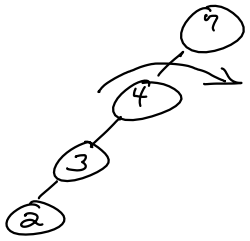


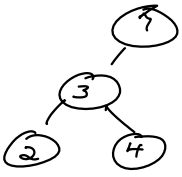


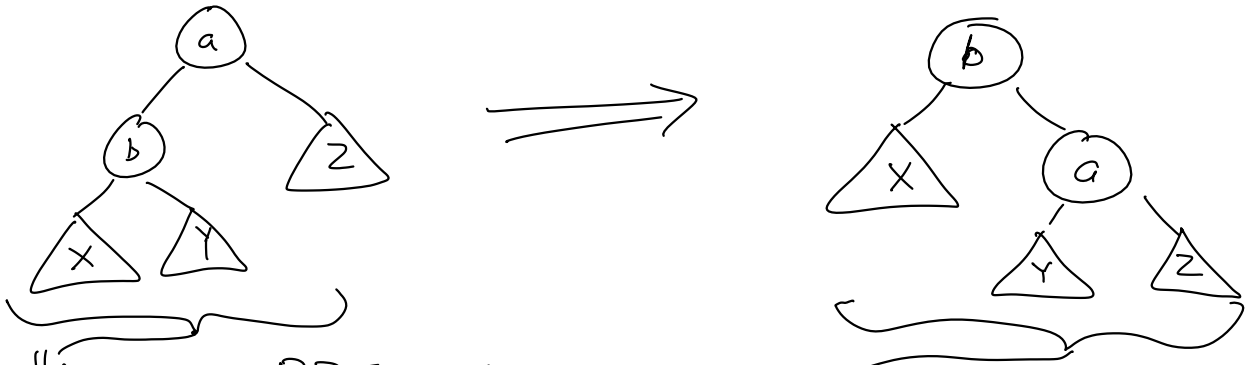












if this is a BST, then why is this a BST?

$a > b$

$a > Y > b$

$X < b$

$Z > a$

Method insertInSubtree(node, key, value):

If node == null:

node ← new Node(key, value)

Return node

Else If key < node.key:

node.left ← insertInSubtree(node.left, key, value)

Return node

Else ....

recalculate height

rebalance