

~~x.left = y.right
if y.right ≠ NIL~~

1. RIGHT-ROTATE(T, x)

$y = x.\text{left}$

$x.\text{left} = y.\text{right}$

if ($y.\text{right} \neq \text{NIL}$)

$y.\text{right}.p = x$

$y.p = x.\uparrow$

if $x.p == \text{NIL}$

$T.\text{root} = y$

else if $x == x.p.\text{right}$

$x.p.\text{right} = y$

else

$x.p.\text{left} = y$

~~$y.\text{right} = x$~~

$x.\uparrow = y$

7

2. INSERT(node, key)

if (node == NIL)
 return Node(key) // approach new node's from key-on

else if (key < node.key)
 node.left = INSERT(node.left, key)

else if (key > node.key)
else
 return node

node.h = 1 + MAX(node.left.h, node.right.h)

balance = node.balance

if (balance > 1 and key < node.left.key)
 return RIGHT_ROTATE(node)

if (balance < -1 and key > node.right.key)
 return LEFT_ROTATE(node)

if (balance > 1 and key > node.left.key)
 ~~LEFTROTATE~~
 node.left = LEFT_ROTATE(node.left)

 return RIGHT_ROTATE(node)

if (balance < -1 and node.right.key)
 node.right = RIGHT_ROTATE(node.right)

 return LEFT_ROTATE(node.right)

return node