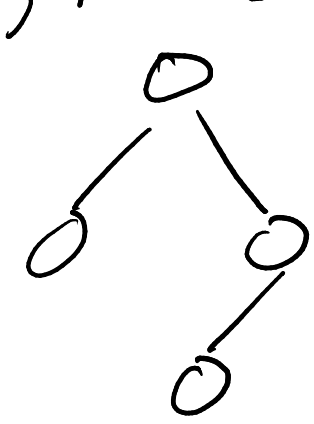
AVL trees (a particular kind of BSTs)
- automatically balancing BSTs

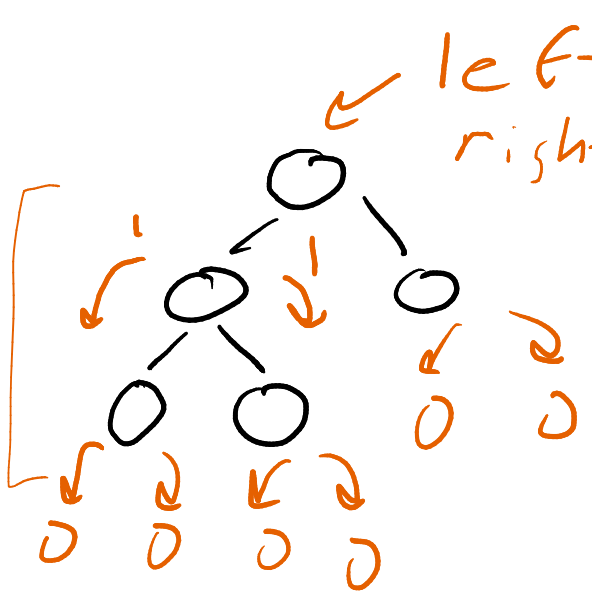
An AVL tree is a BST where at every node, the heights of the left and right subtrees differ by at most 1

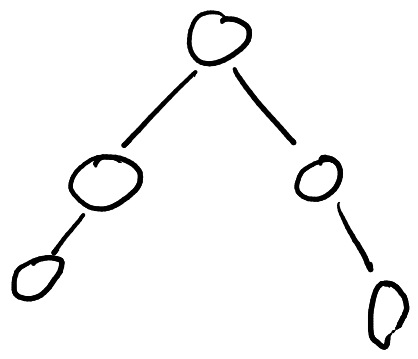


T
3
I

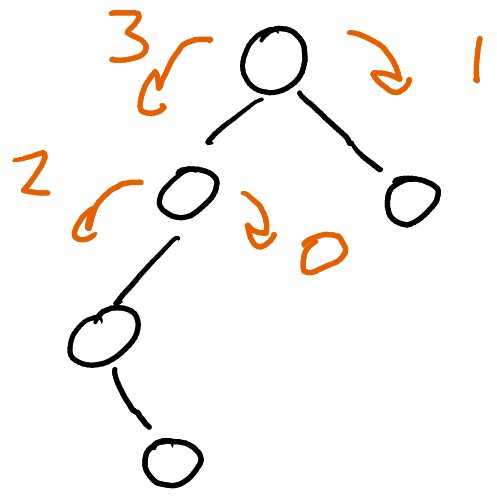
→ max number of nodes on a path from root to leaf

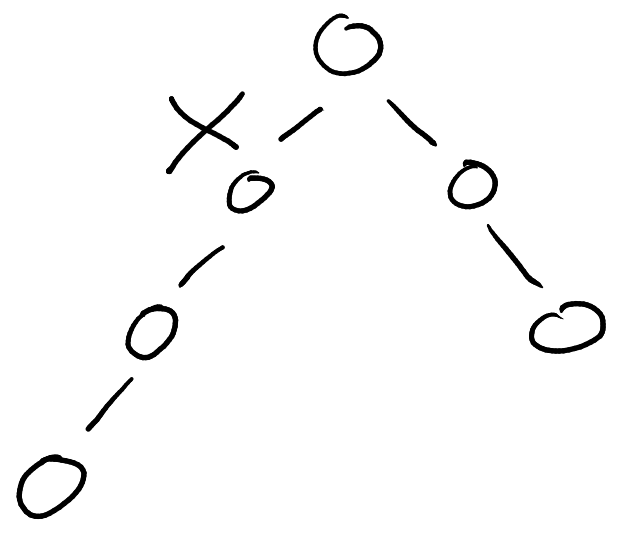
AVL tree examples





Not AVL trees



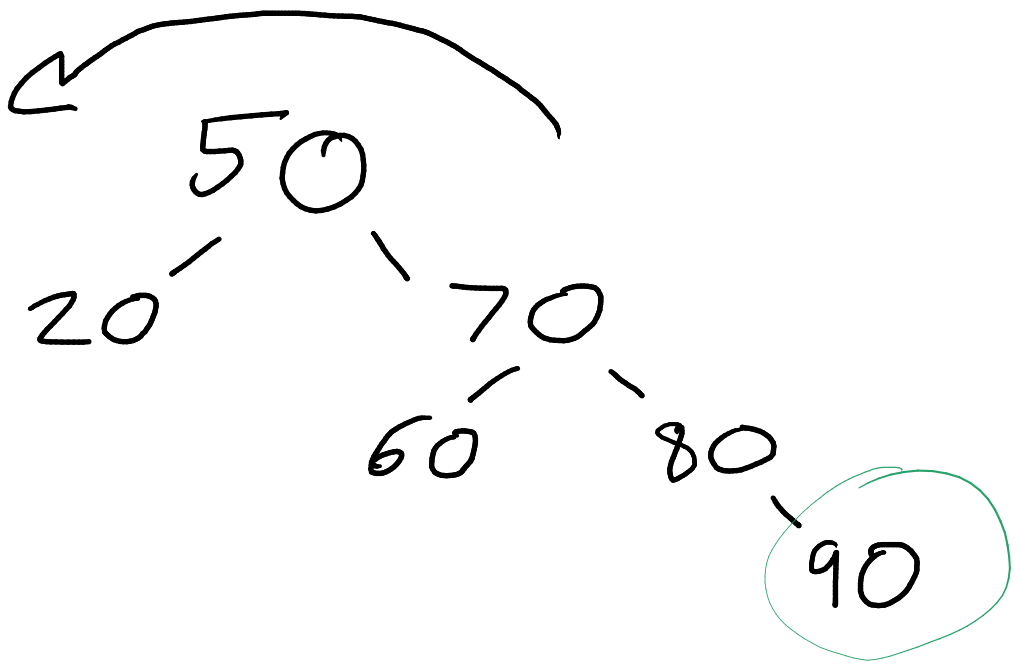


(Reminder: BST: all nodes on left have values < node, all nodes on right have values > node)

Worst case, an AVL tree has height $\simeq 1.44 \log_2 n$

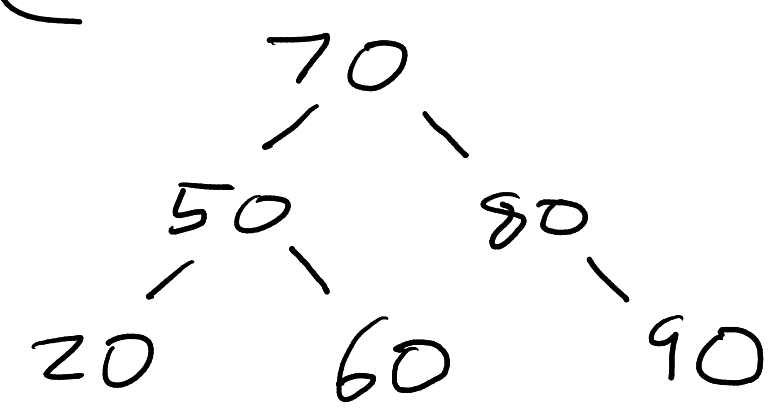
How do you insert?

First attempt - inset as usual for a BST, and if AVL tree balance is ok, no worries.



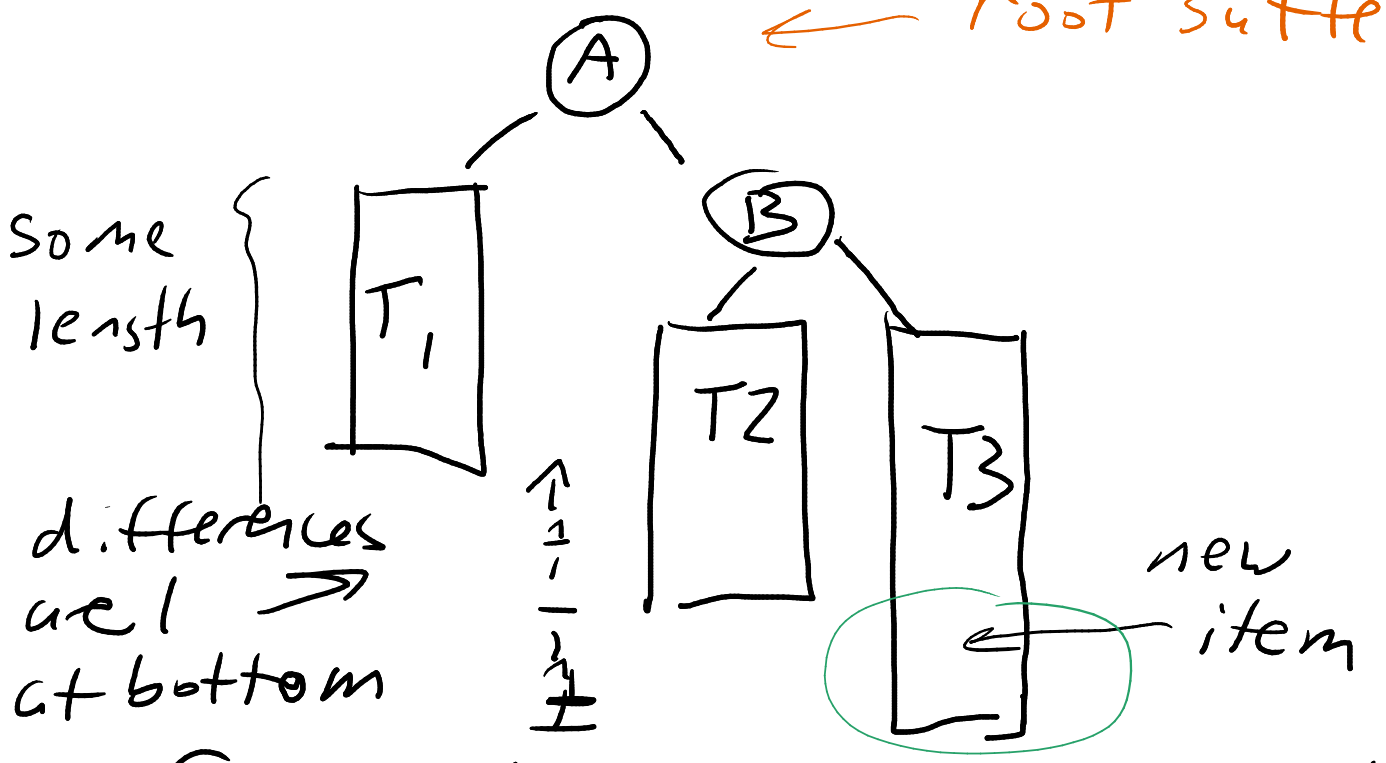
Insert 50
20
70
60
80
90

50 is now unhappy.
In this specific case, fix it by a left rotation around the broken 50 node



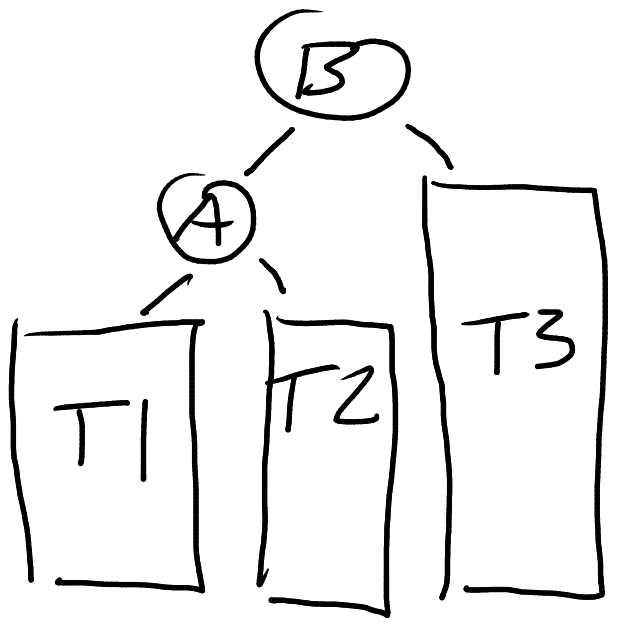
change of allegiance
(it jumped to the 50)

Generally, this case looks like:

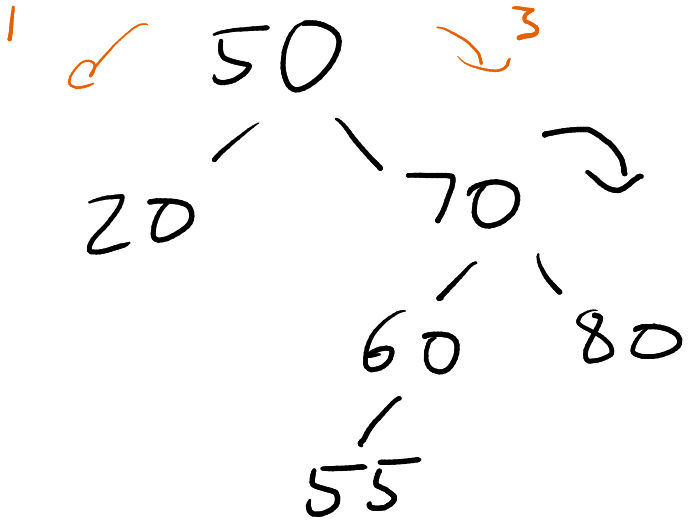


← root suffers: left side is 2 less than right side and the problem subtree is all the way on the right outside

fix w/ a left rotation

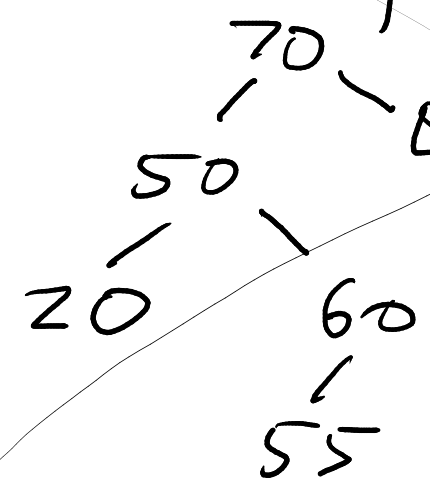


There's a mirror image where left side is too deep and you do a right rotation



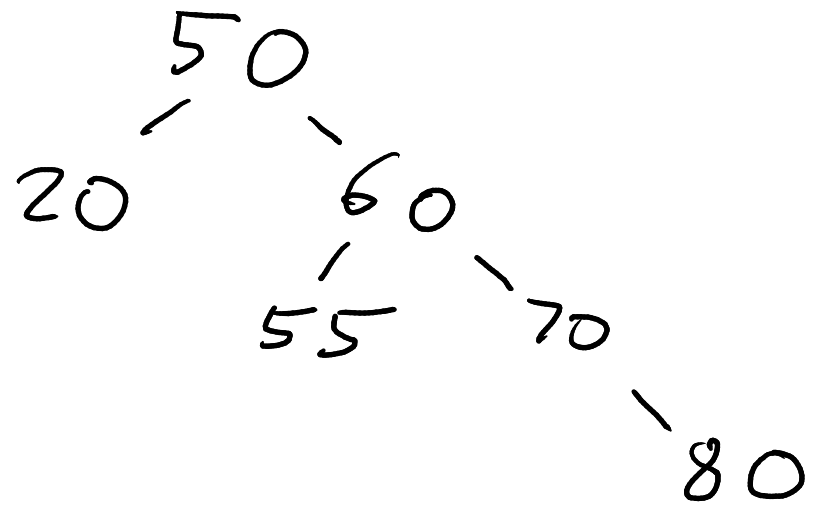
50
20
70
60
80
55

Left rotation doesn't work. Let's see what happens.

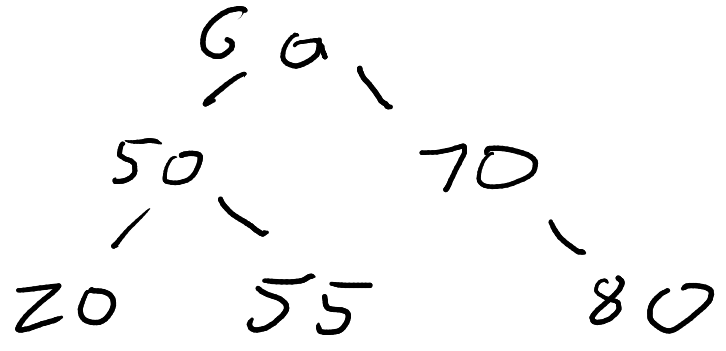


Instead, need 2 rotations.

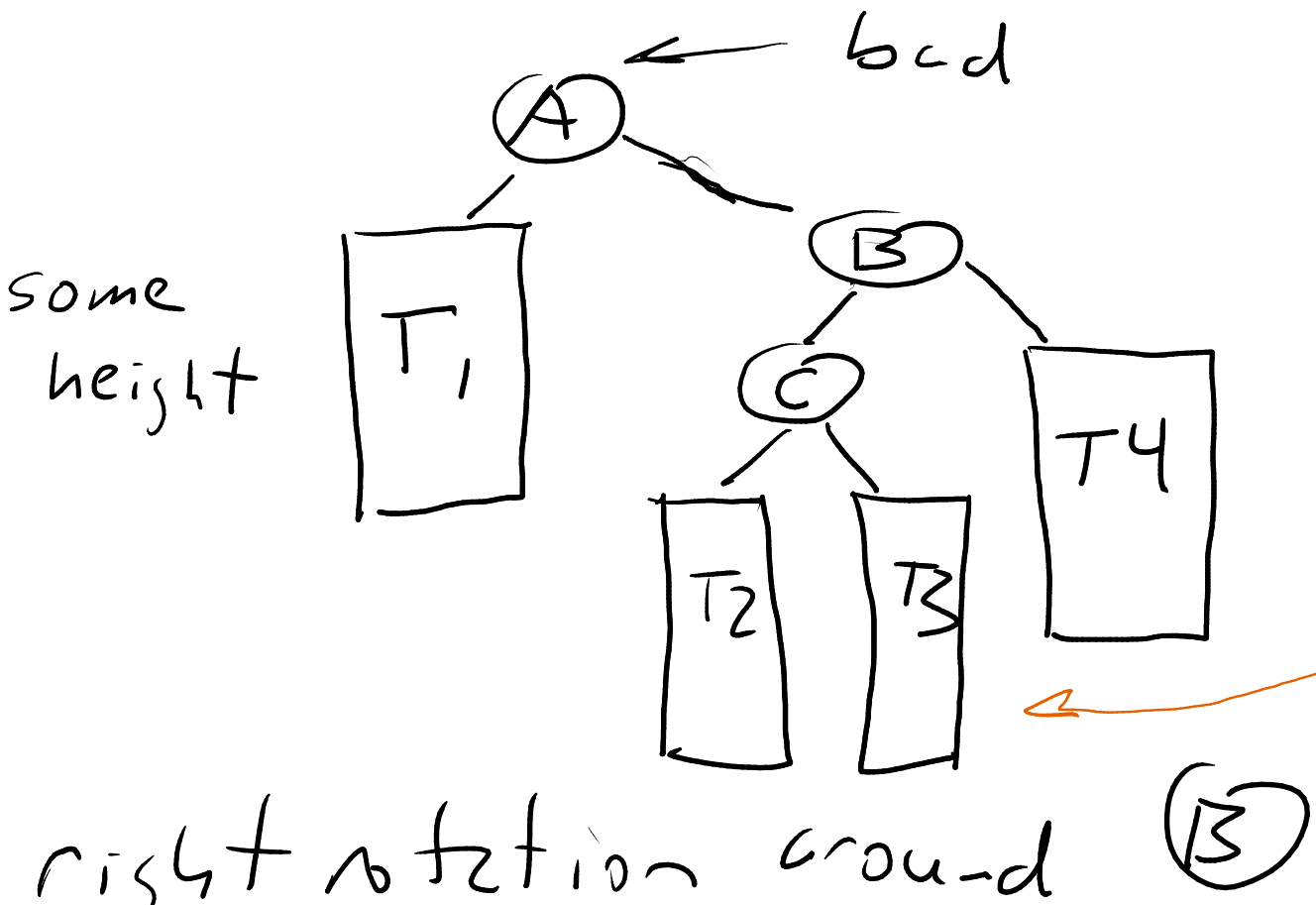
First, rotate right around 70

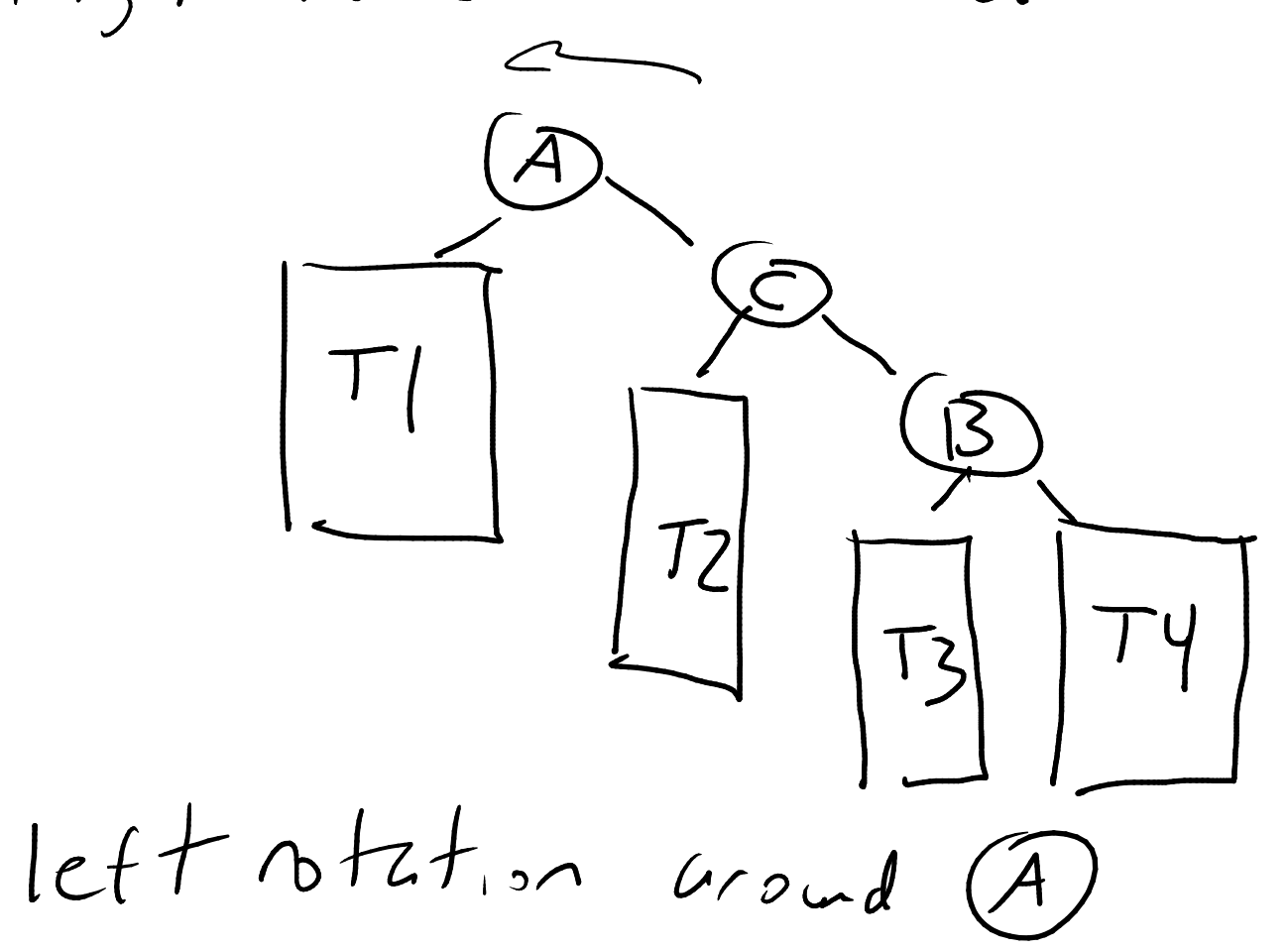


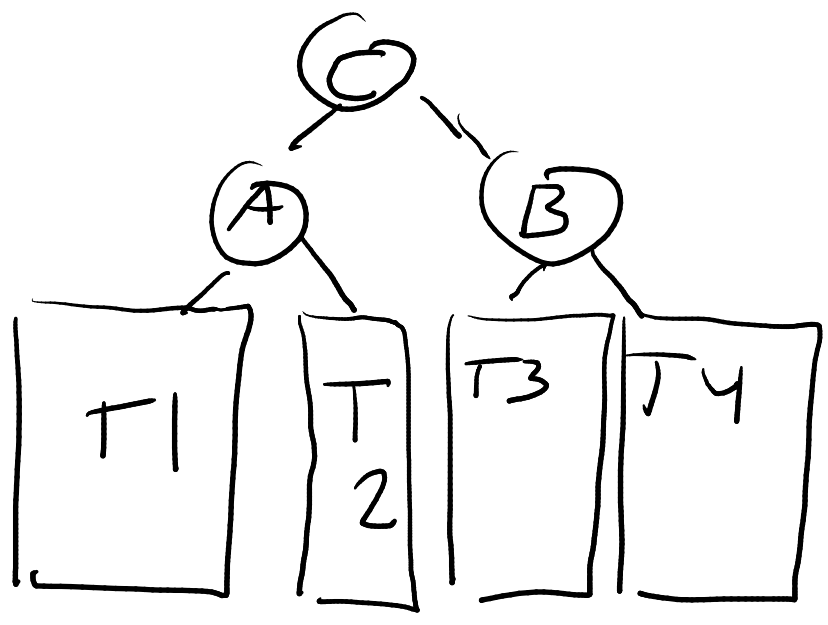
Then rotate left around 50



General case that results in a right rotation, followed by a left

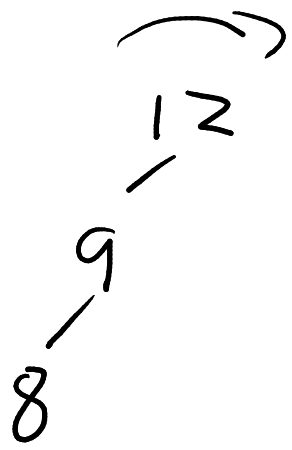


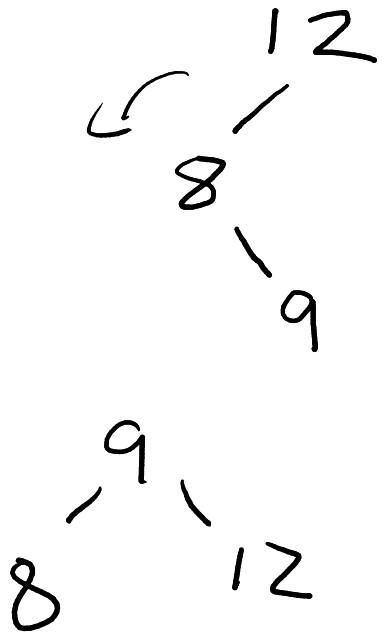




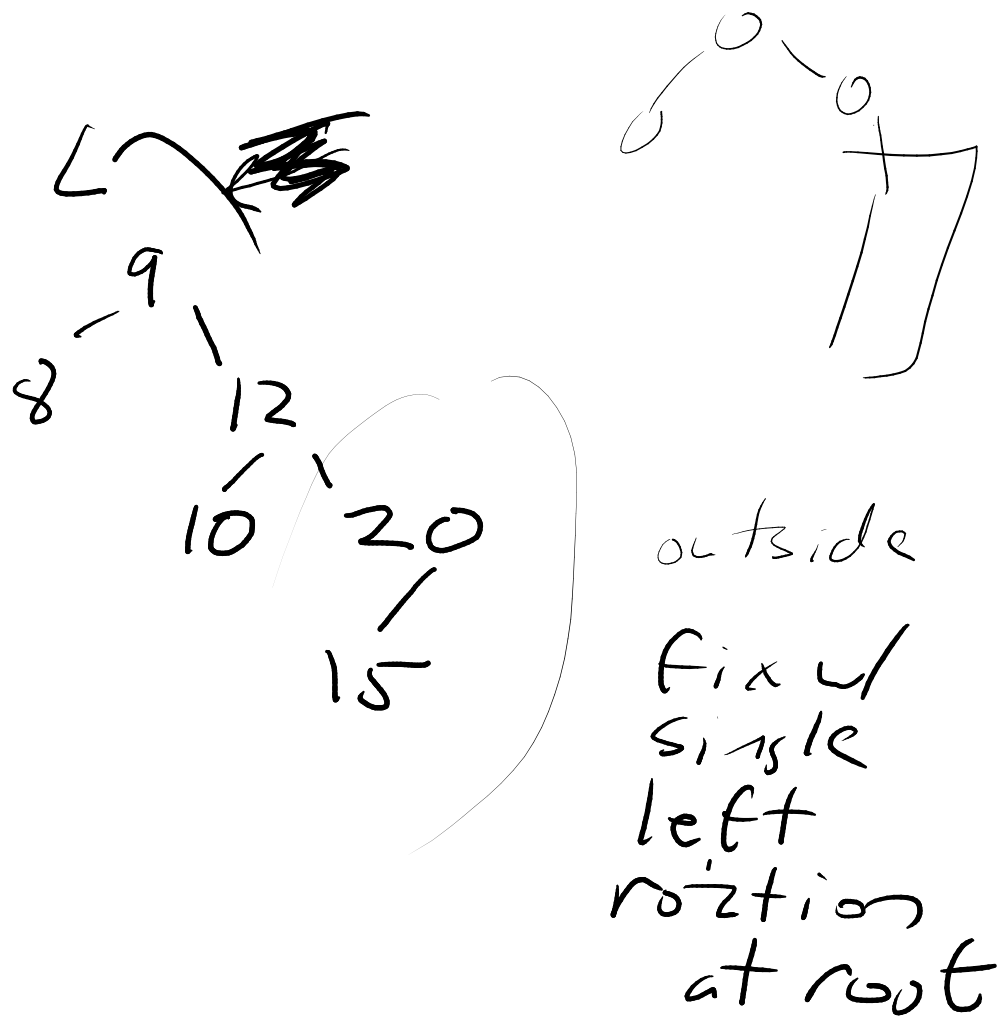
Likewise, there is a mirror image

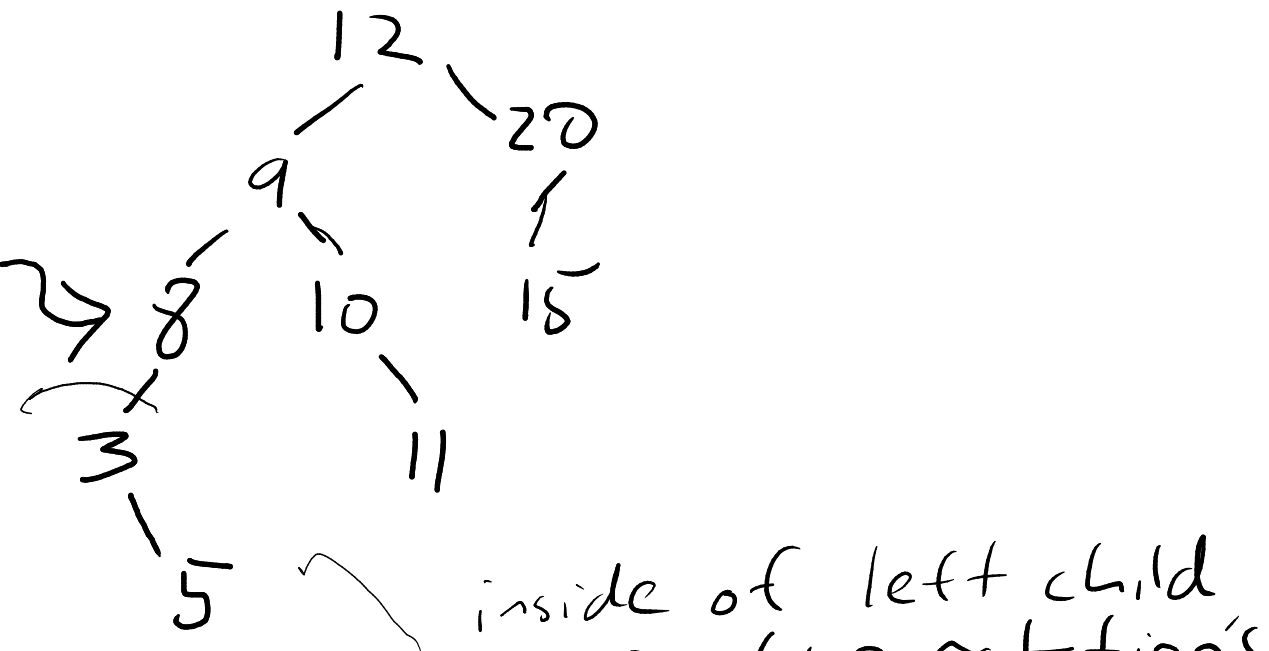
Worksheet

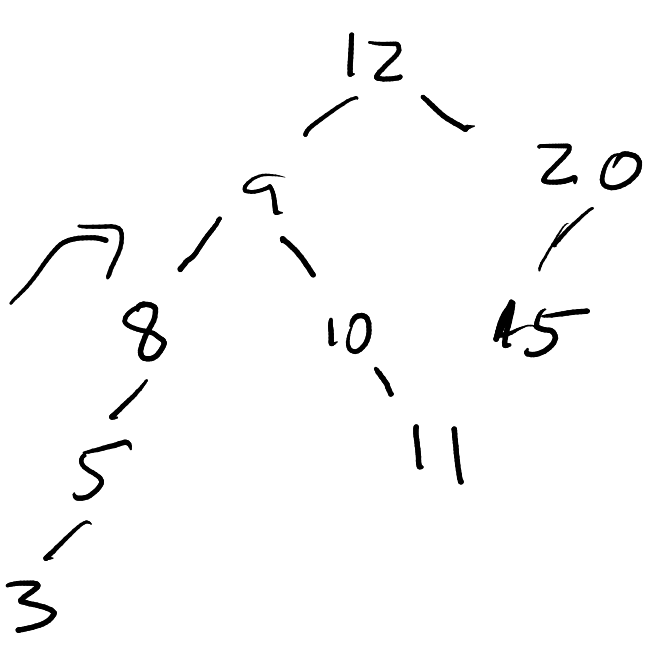


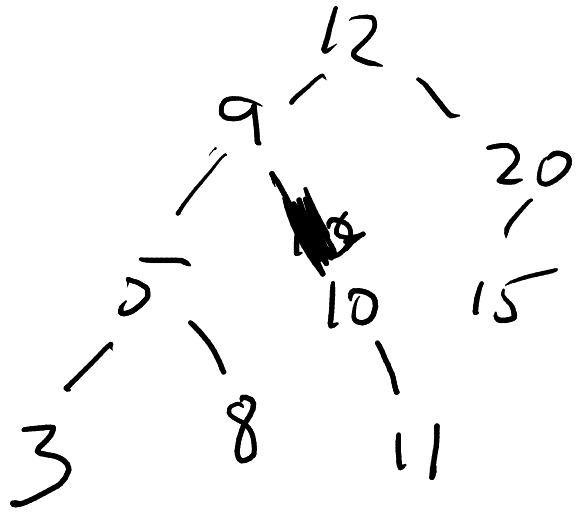


left rotation, then right



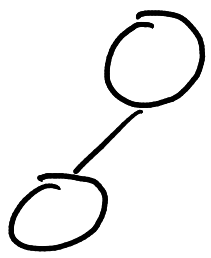






Minimum nodes for $h = 1$

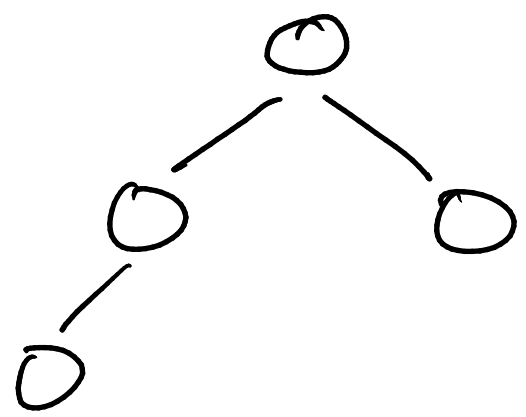
Minimum nodes for $h = 2$



In general minimum nodes for $h = n$ is an AVL tree with minimum $n-1$ on one side and minimum $n-2$ on the other

Minimum nodes for $h = 3$

$h = 2$



$h = 1$