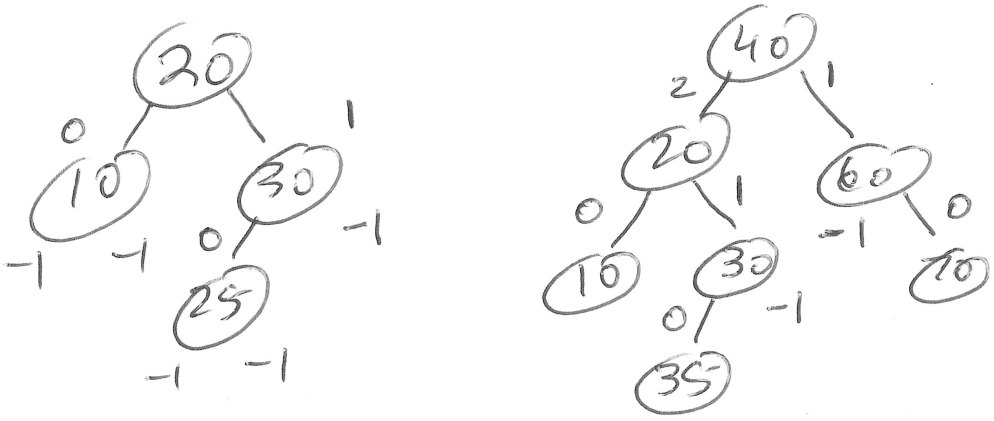
COP 3502 3/5/24

AVL Trees

- Binary Search Tree
- each node follow the avl tree node property

abs(height(left subtree) - height(right subtree)) ≤ 1



Any valid AVL tree w/ n nodes has height O(lg n).

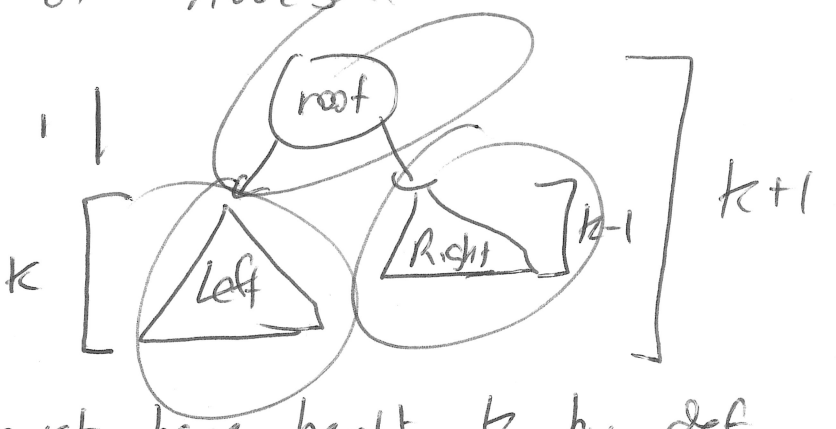
The least number of nodes of an AVL tree of height h is $F_{h+3} - 1$. (F_k is the k^{th} Fib #)

h=0, (20) → 1 node, $F_{0+3} - 1 = 2 - 1 = 1$ ✓

h=1, (20)(30) → 2 nodes, $F_{1+3} - 1 = 3 - 1 = 2$ ✓

1. H assume true for all non-neg int h, $h \leq k$, k is a pos int, arbitrarily chosen ≥ 1. $T(h) = F_{h+3} - 1$.
1. S Prove for $h = k+1$ that the fewest # nodes in an AVL tree of height $k+1$ is $F_{k+4} - 1$.

Imagine an AVL tree of height k+1 w/ the minimal # of nodes:



max subtree must have height k by def.
other subtree must be at least height k-1 by AVL tree property. (Let $T(k)$ = min # of nodes in AVL tree of height k.)

$$T(k+1) = T(k) + T(k-1) + 1$$

$$= F_{k+3} - 1 + F_{k+2} - 1 + 1$$

$$= \boxed{F_{k+4} - 1}$$

a:b equal
b:a+b

$$T(h) = F_{h+3} - 1$$

$\log_\phi 2 \geq 1.?$

node $\leftarrow$ $n \geq F_{h+3} - 1$

Golden ratio ϕ

$$F_{h+3} \leq n+1$$

$$\frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^{h+3} - \left(\frac{1-\sqrt{5}}{2}\right)^{h+3}\right) \leq n+1$$

$$\frac{1}{\sqrt{5}}\phi^{h+3} \leq n+1$$

$$\phi^{h+3} \leq c(n+1)$$

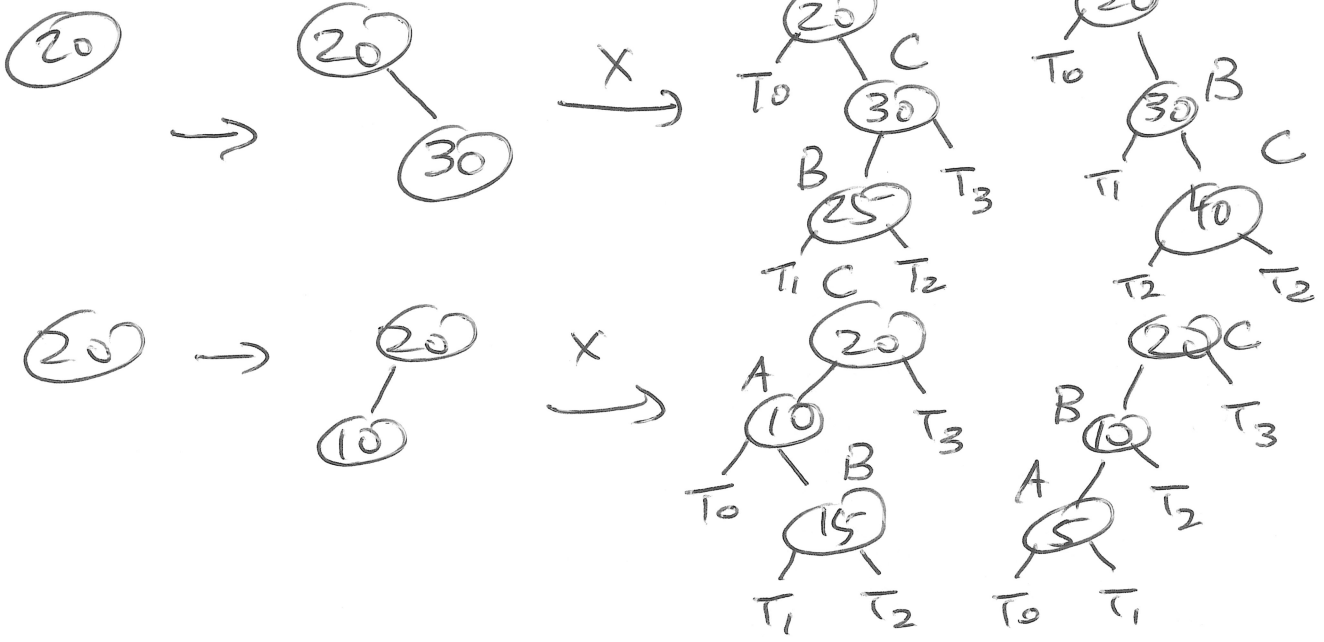
$$h+3 \leq \log_\phi(c(n+1))$$

$$h \leq \log_\phi c(n+1) - 3$$

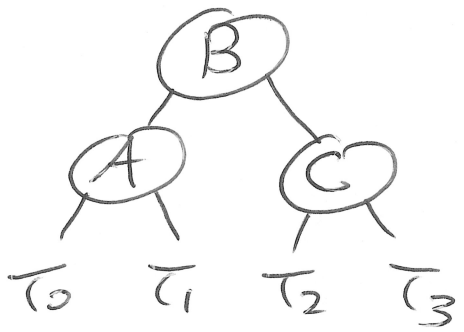
$$\leq c\log_\phi(n)$$

How do we maintain?

Insert



FIX 4 : FOR ALL



Insert Recursively

root->left = insert(...
look at new height

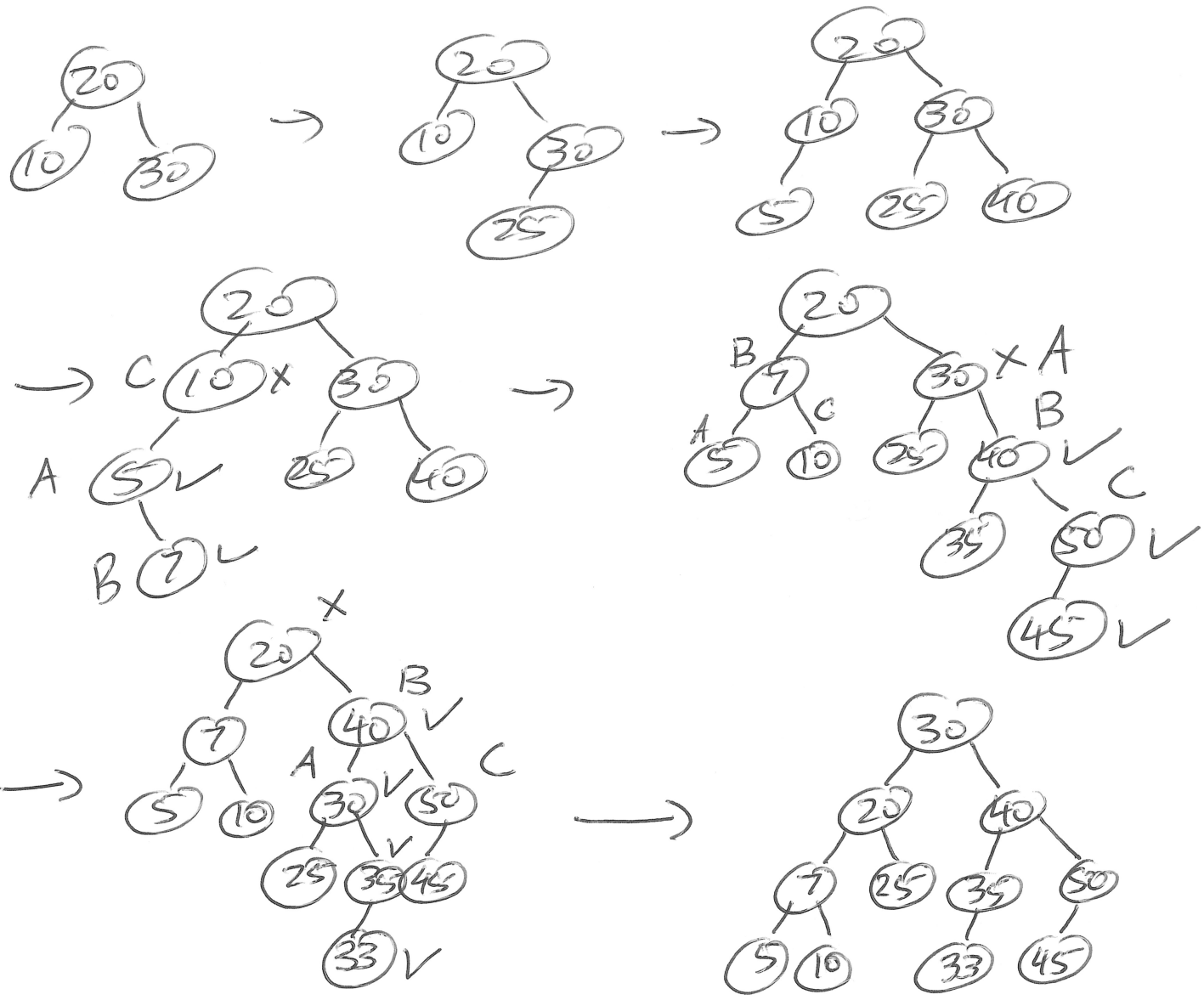
a) Do regular bst code

b) update left + right heights (only 1 of these will be updated)

c) update my height

d) If there's an imbalance, fix it.

In code I check this each time, but it never gets called more than once



Delete

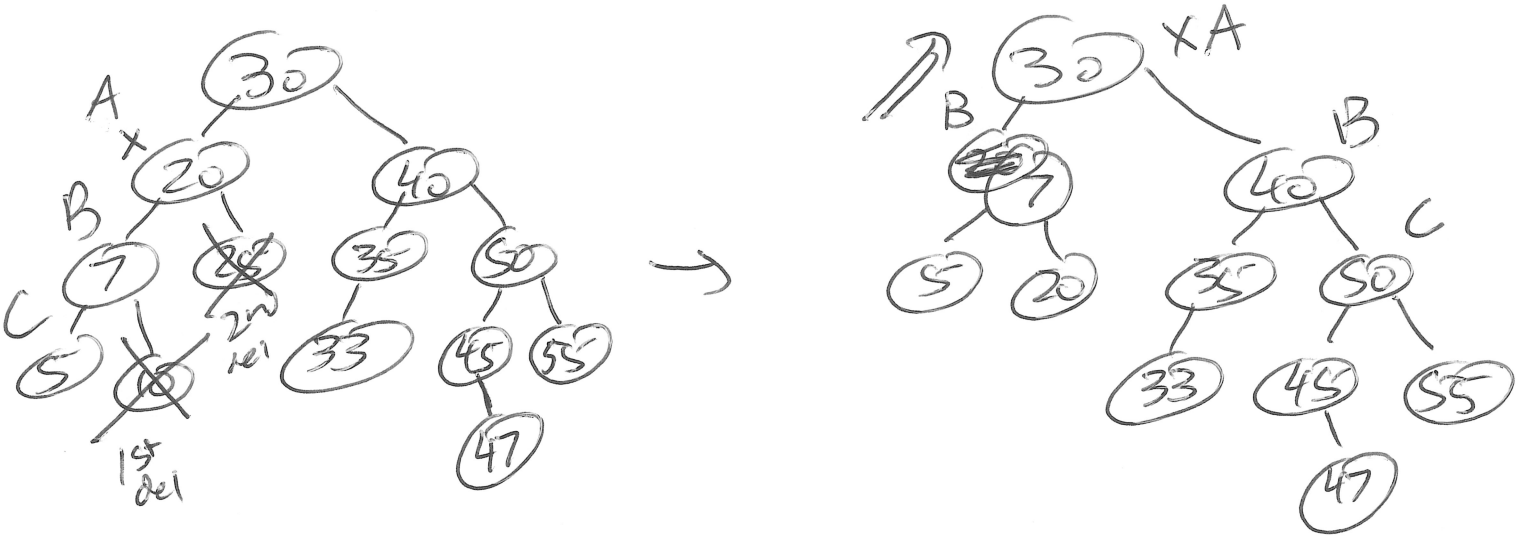
Same exact structural code as insert

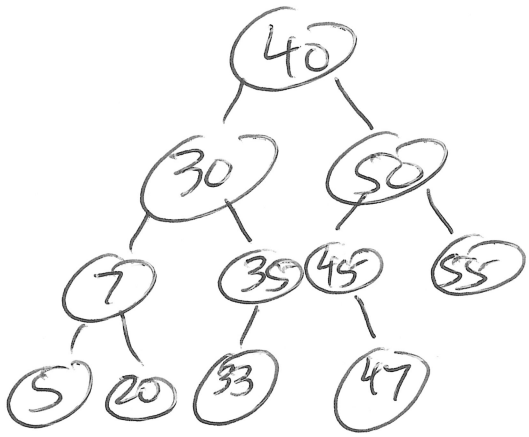
(a) do regular delete

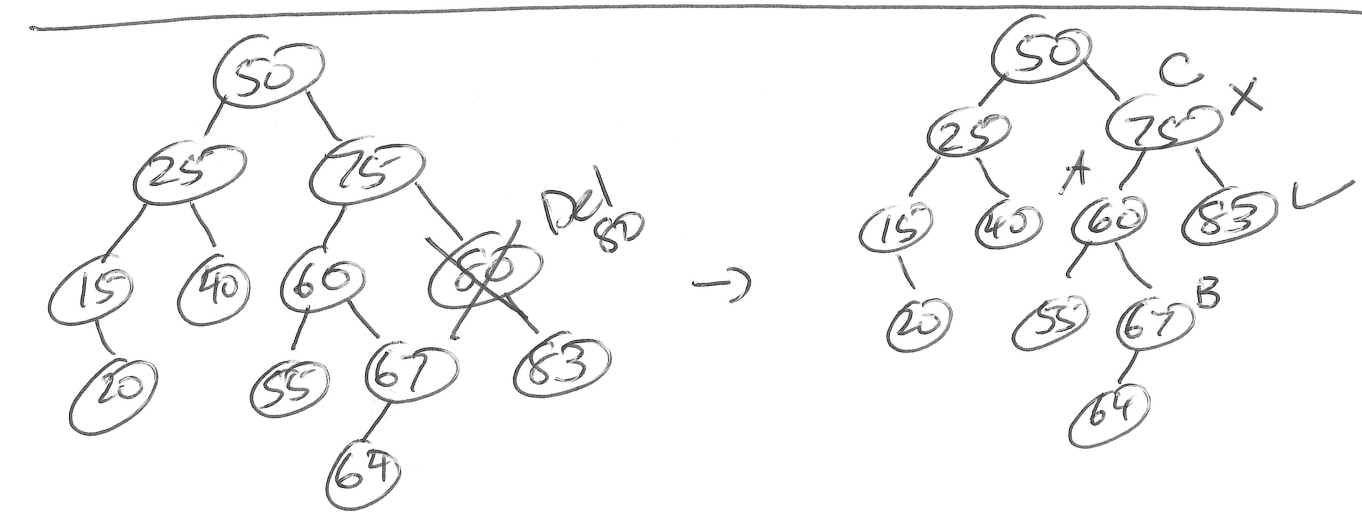
(b) update height

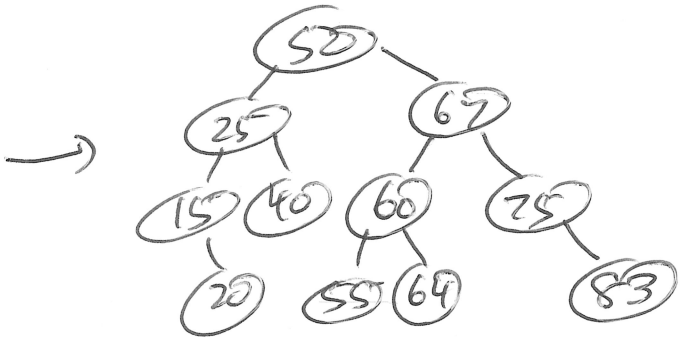
(c) if imbalance, restructure ⇒ might trigger more than once!

(d) return new root


A
X
30
20
40
B
7
35
50
C
5
33
45
55
47
1st del
2nd del
XA
B
B
C


40
30
50
7
35
45
55
5
20
33
47


50
25
75
15
40
60
80
20
55
67
83
64
Del 80
C
X
A
B
L


50
25
67
15
40
60
75
20
55
64
83