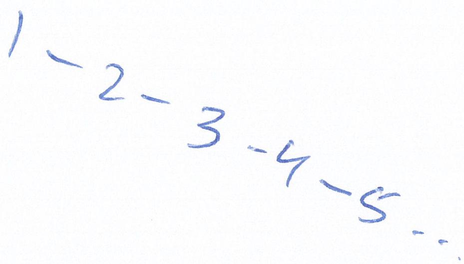
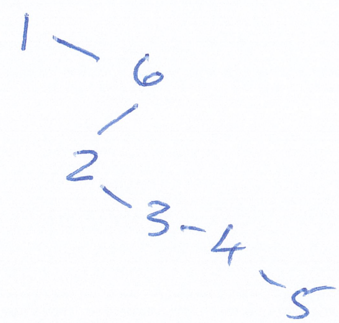
COP 3502 3/31/2026

AVL Trees

Issue w/ regular BST is search, insert, delete run in $O(h)$ time, h = height, WORST CASE $h = n-1$, n = # Nodes





2^{n-1} BSTs of n ints of height $n-1$.

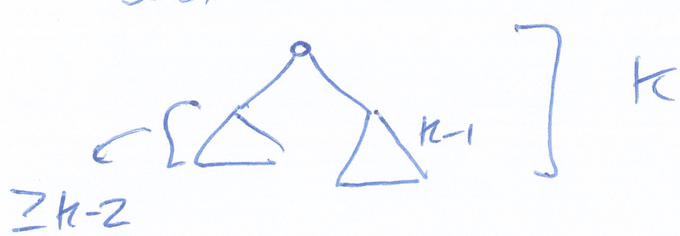
WORST CASE RUN-TIME IS $O(n)$ ops

AVG CASE IS $O(\lg n)$

Goal = guarantee $O(\lg n)$ height worst case

AVL Tree Node Property

1) regular BST (left lower, right higher)
2) for every node, the height of the left subtree + height of right subtree can't differ by more than 1.



Let T_h = min number of nodes for an AVL tree of height h.

Prove $T_h = F_{h+3} - 1$, where F_n is the n^{th} Fibonacci #.

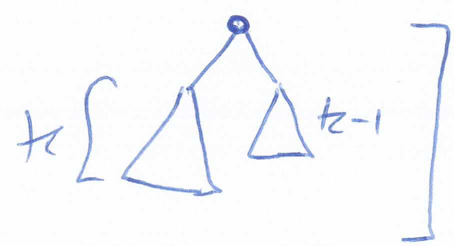
$h = 0$ $\circ$, $T_0 = 1$ $LHS = T_0 = 1$, $RHS = F_{0+3} - 1 = F_3 - 1 = 2 - 1 = 1$ ✓

$h = 1$, $T_1 = 2$ $LHS = T_1 = 2$, $RHS = F_{1+3} - 1 = F_4 - 1 = 3 - 1 = 2$ ✓

I.H. Assume for all non-neg int n, $n \leq k$ that $T_n = F_{n+3} - 1$ where k is an arbitrarily chosen positive integer.

I.S. Prove $\cancel{T_{n+1}} = T_{k+1} = F_{k+4} - 1$.

$$T_{k+1} = T_k + T_{k-1} + \underbrace{1}_{\text{root}}$$



$$= F_{k+3} - 1 + F_{k+2} - 1 + 1, \text{ using I.H.}$$

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

n = # nodes, h = height

$$n \geq F_{h+3} - 1 \qquad F_n = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$

$$n + 1 \geq F_{h+3} \qquad F_n \sim c\phi^n$$

$$n + 1 \geq c \cdot \phi^{h+3}$$

$$\phi^{h+3} \leq \frac{n+1}{c} \rightarrow h + 3 \leq \log_\phi \frac{n+1}{c}$$

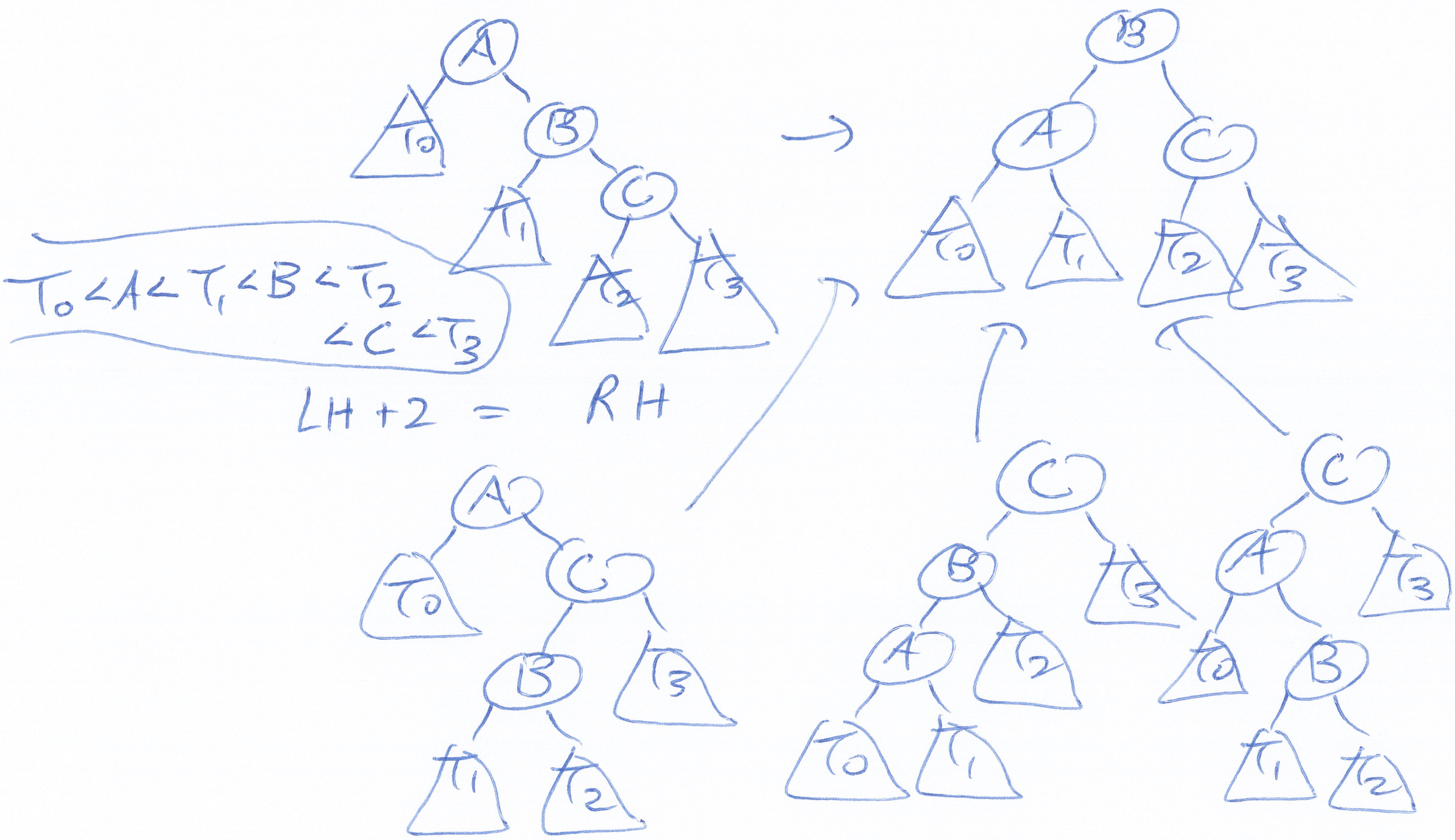
$$h + 3 \leq O(\lg n)$$

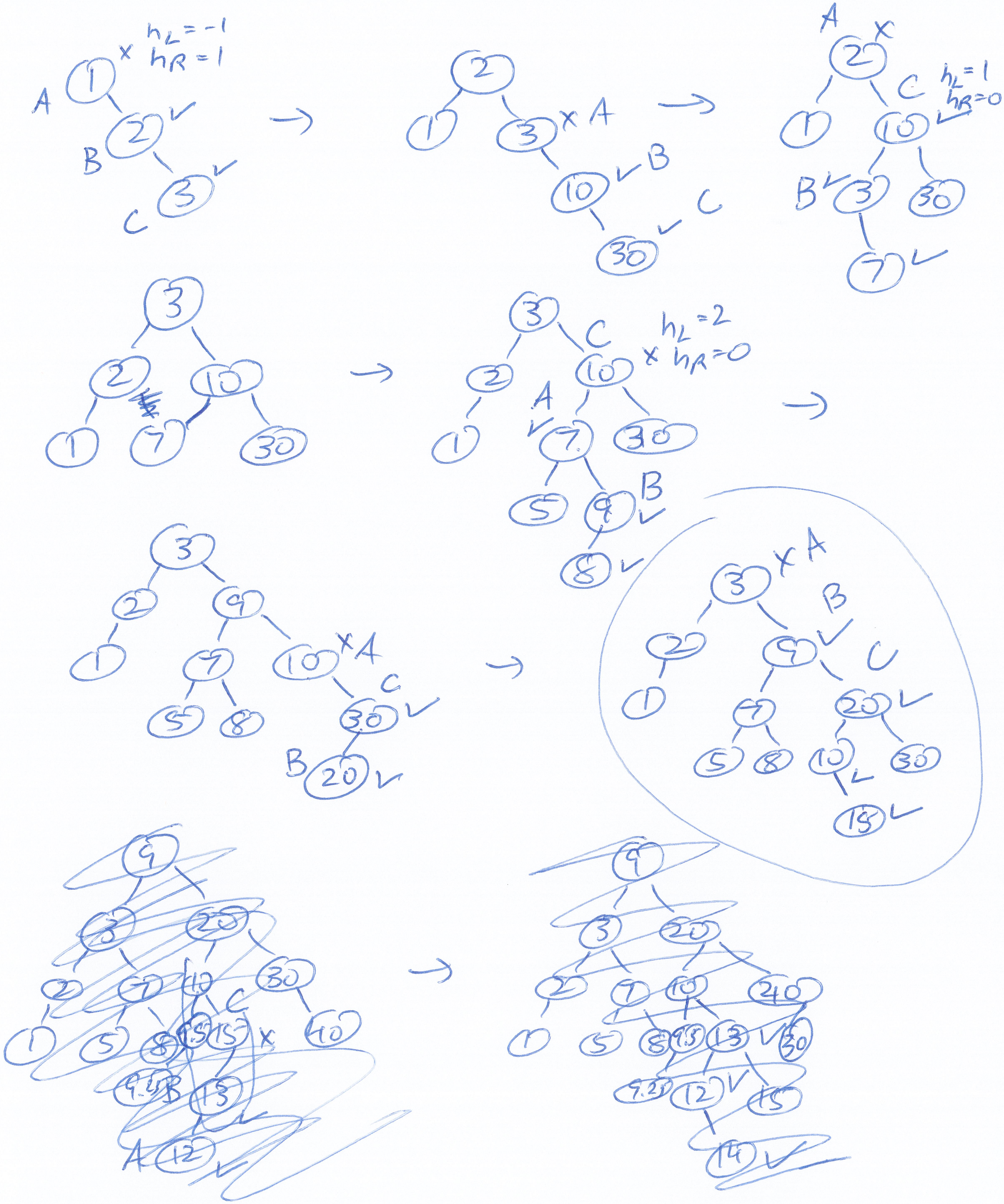
AVL

Insert - Step 1 - regular bst insert

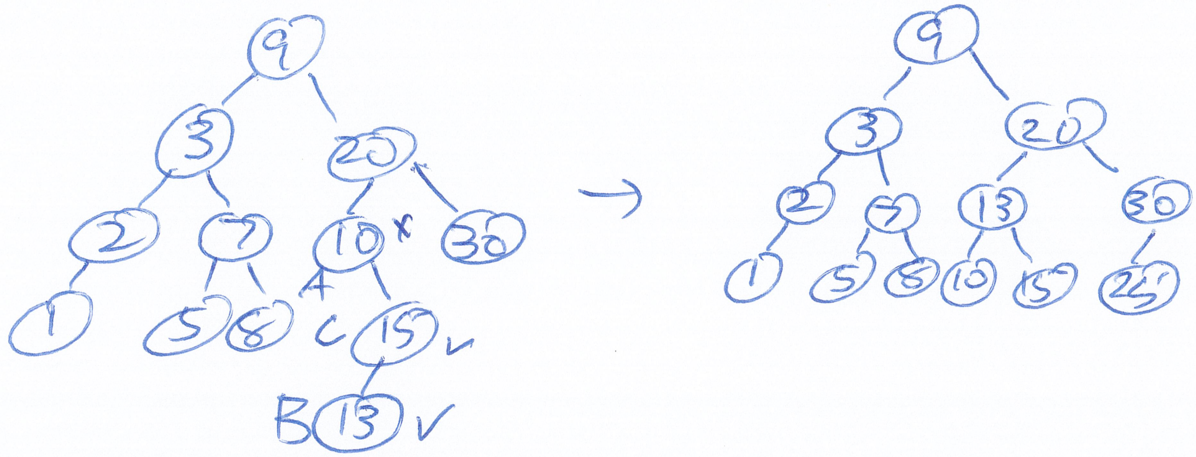
Step 2 - trace up the ancestral path and if there's an imbalance, fix it!

General Procedure to Fix





hL = -1
hR = 1
hL = 1
hR = 0
hL = 2
hR = 0



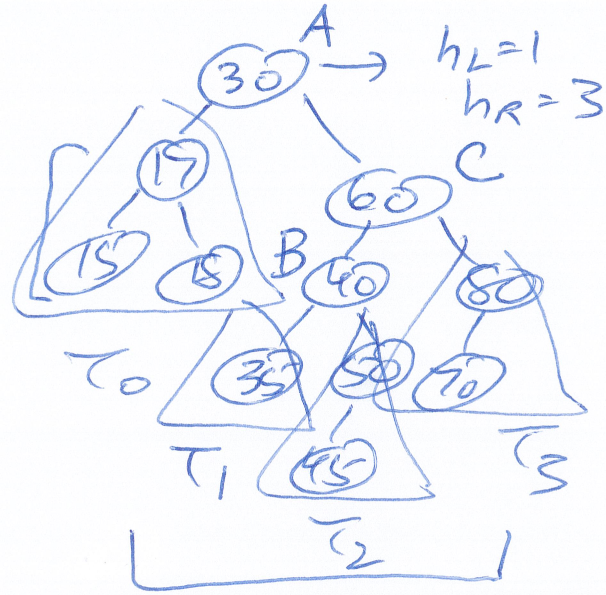
Delete

1) Do a regular BST Delete

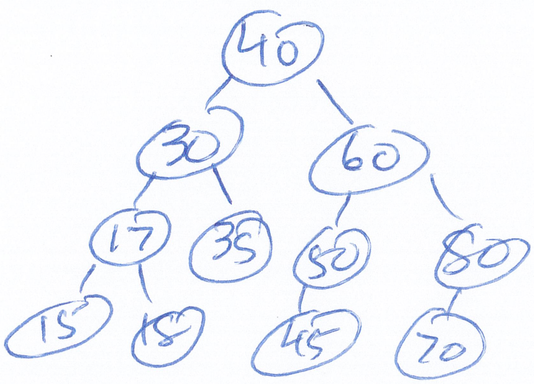
2) Trace up ancestral path of ~~paren~~ parent of deleted node and rebalance whenever there's a problem

→ Can trigger at each node it traces up.

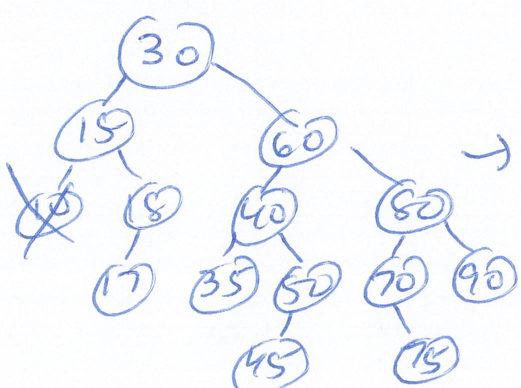
Delete 10



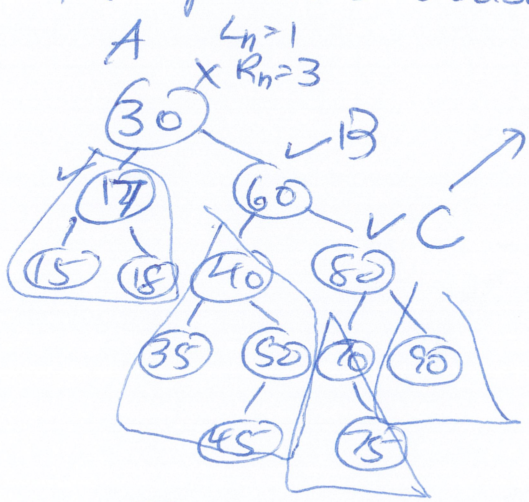
we choose C, B due to greater height



Note: I was right: when choosing A, B, C for delete you go to the longer subtree. If the 2 sides are equal, you ALWAYS go in the same direction as the previous decision. Here is a case

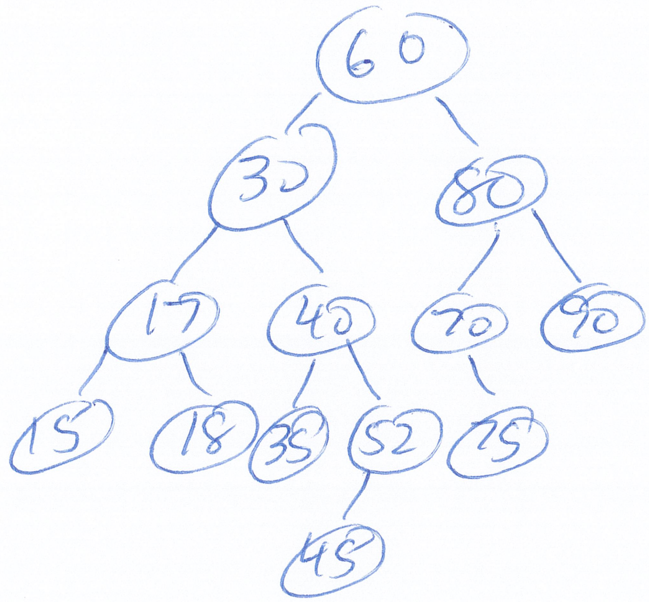


Del 10



we choose this because 60 is right of 30, so we went Right again in the tie case

Final answer on next page


60
30
80
17
40
70
90
15
18
35
52
75
45