COP 3502 10/29/25

1) Considering syllabus change - will vote in class on Friday.

2) Updated Grade Calc - Tomorrow Morning

3) AVL Trees

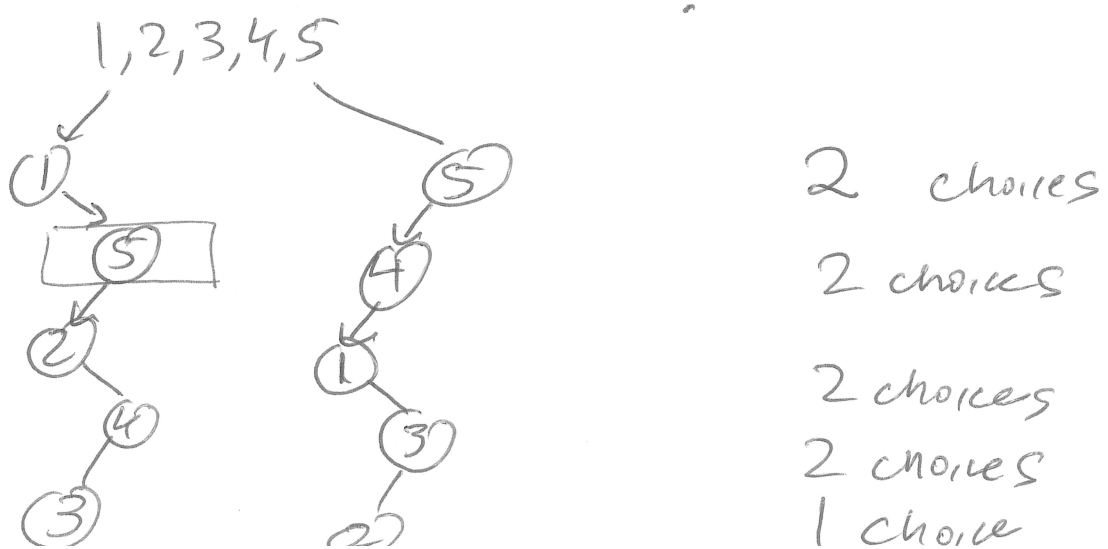
- Specific Type of Binary Search Tree

Regular BST

Positives: inserts, deletes, searches - $O(h)$ time
h = height and ON AVERAGE $h = O(\lg n)$,
where n = # of elements

Negative: WORST CASE is $h = n-1$, and these run times are $O(n)$.

# of BSTs storing 1 to n with n nodes + height n-1 is 2^{n-1}.



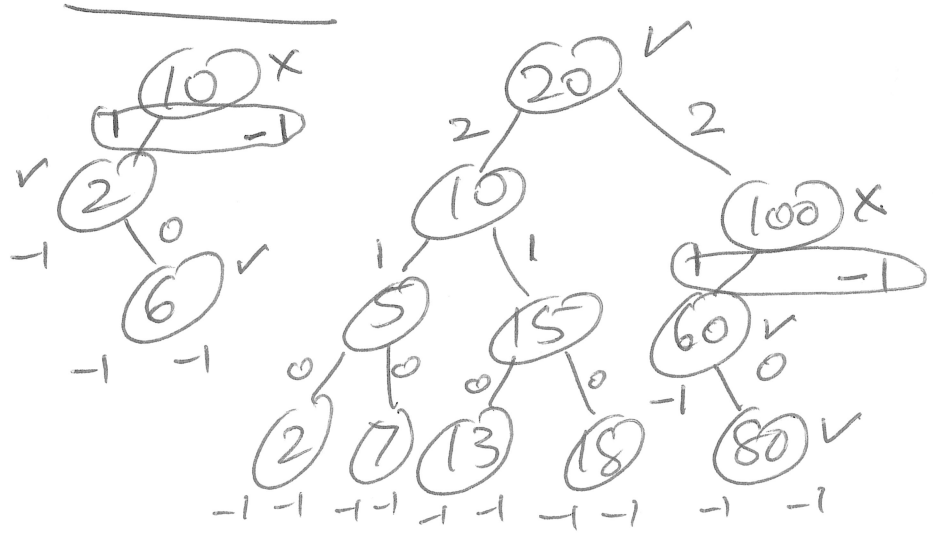
AVL Tree - Solution to height to # node ratio guaranteed

(1) Regular BST Property

(2) AVL Tree Node Property:
for any node in the tree,
the difference in height (absolute value)
of the left subtree and right subtree
is ≤ 1.

Valid

Invalid



2 Things

1) Prove that a tree with this property has height $O(\lg n)$

2) How to maintain these properties

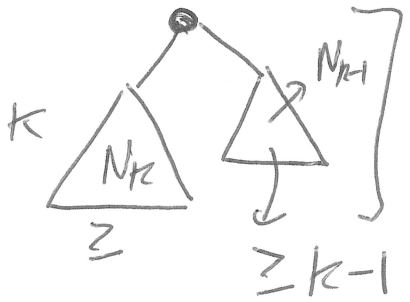
→ Given a tree of height h, we'll prove n, the number of nodes in tree is $\geq F_{h+3} - 1$ where F_k denotes the k^{th} Fibonacci #.

$h=0 \quad n=1, \quad \geq F_{0+3} - 1 = 2-1 = 1$ ✓

$h=1 \quad \underset{(2,3)}{n \geq 2} \quad \geq F_{1+3} - 1 = 3-1 = 2$ ✓

I.H. Assume for all $h \leq k$ that the number of nodes in an AVL tree of height h, $N_h \geq F_{h+3} - 1$.

Prove for $h = k+1$ that $N_{k+1} \geq F_{k+3+1} - 1$.



$N_{k+1} \geq N_k + N_{k-1} + 1$ (root)

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

$= F_{k+4} - 1$

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

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

$$n+1 \geq F_{h+3} \longrightarrow \phi$$

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

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

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

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

$\lg c$ → Constant

$$\lg(n+1) - \lg c \geq (h+3)\lg\phi$$

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

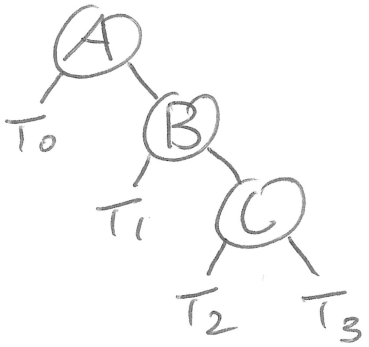
$$h = O(\lg n)$$

AVL Inserts

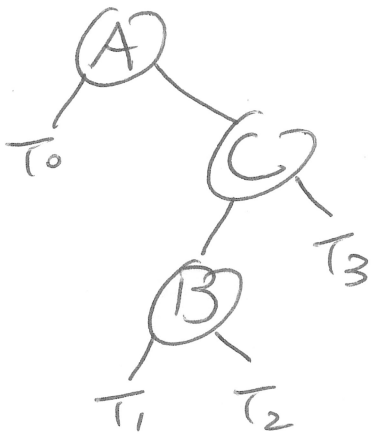
1) Regular BST insert

2) Trace up the ancestral path. If you reach a node with an imbalance, fix it! Then you're done!

Four Types of Imbalances



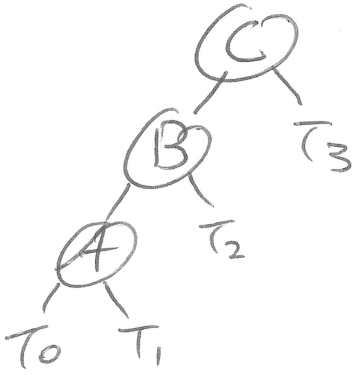
LH < RH-1

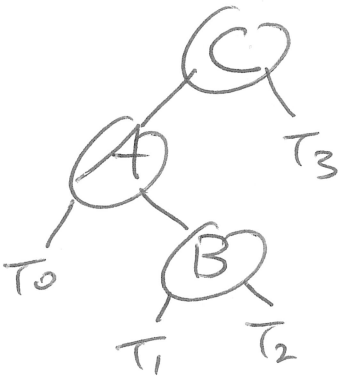


LH < RH-1

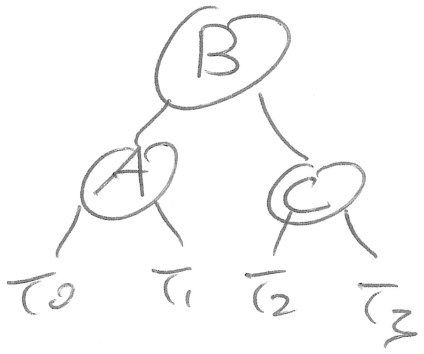
T_0, T_1, T_2, T_3 are subtrees possibly empty

$T_0 < A < T_1 < B < T_2 < C < T_3$



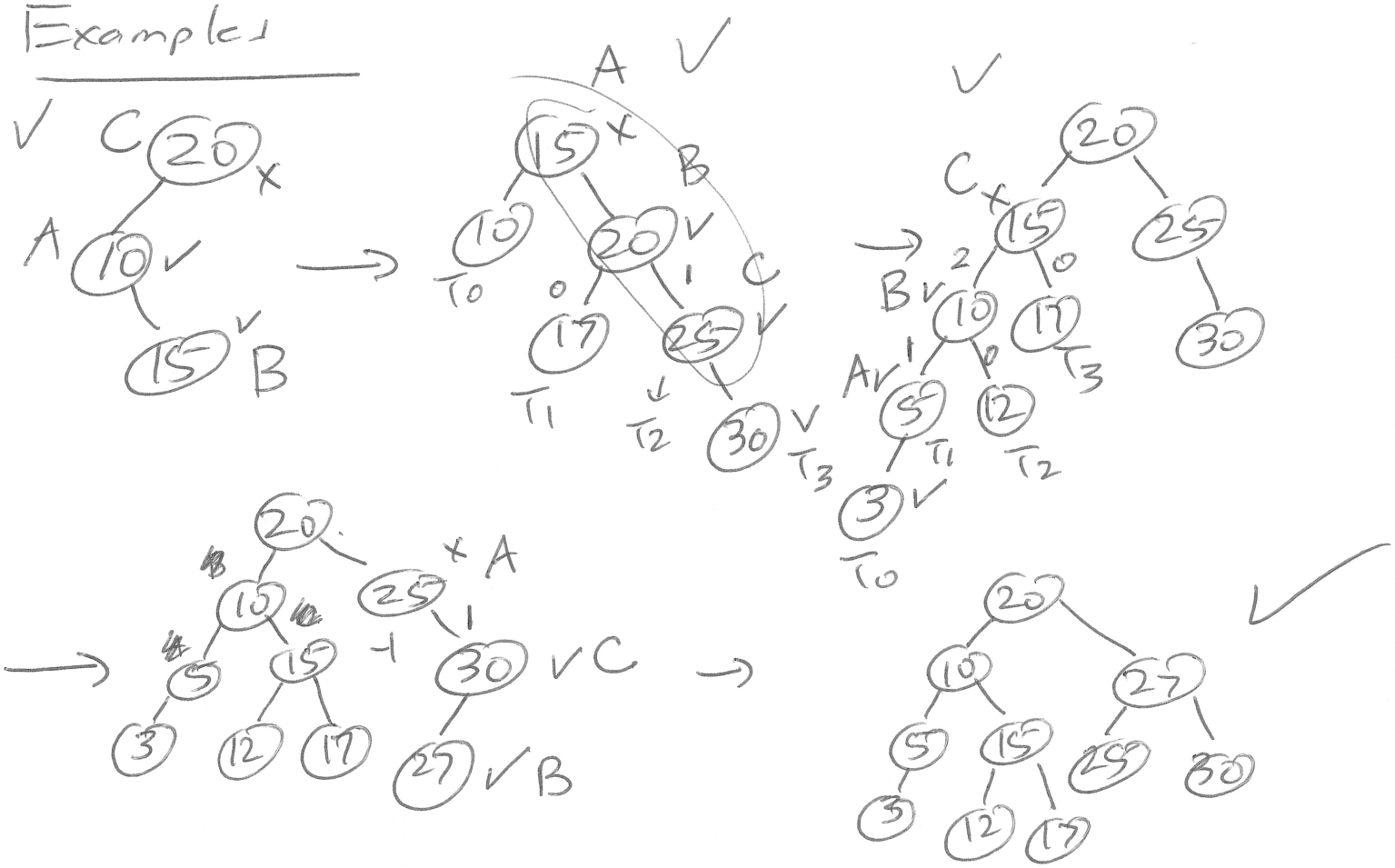


FIX →



RH < LH-1

Example


V
C
20
x
A
10
V
15
V
B
A
V
15
x
B
10
T0
20
V
0
1
C
17
25
V
T1
T2
30
V
T3
V
20
C
x
15
25
2
0
B
V
10
17
30
1
0
T3
A
V
5
12
T1
T2
3
V
T0
20
B
10
25
x A
1
5
15
30
V C
3
12
17
27
V B
20
10
27
5
15
25
30
3
12
17
V