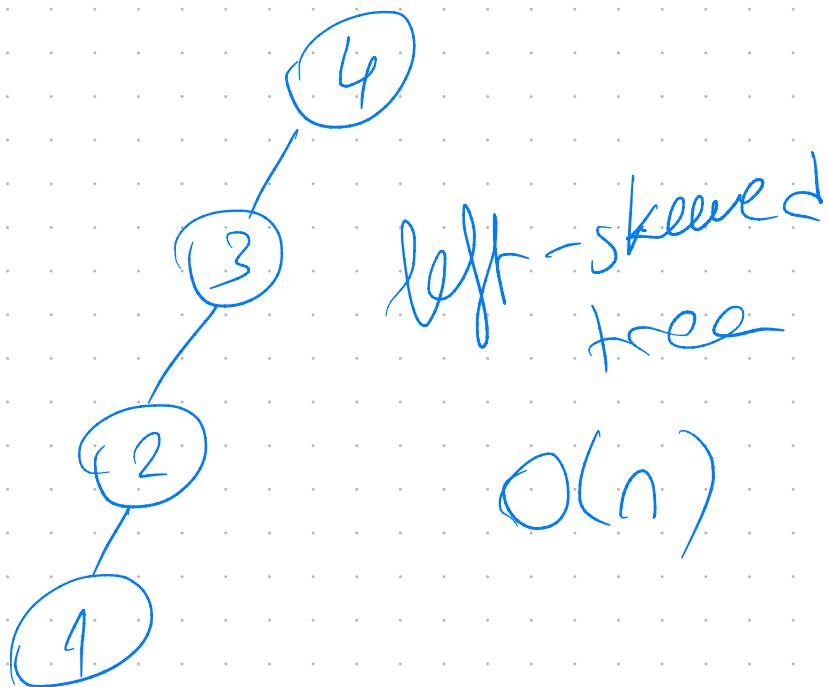


## AVL Trees:

Self balancing BST



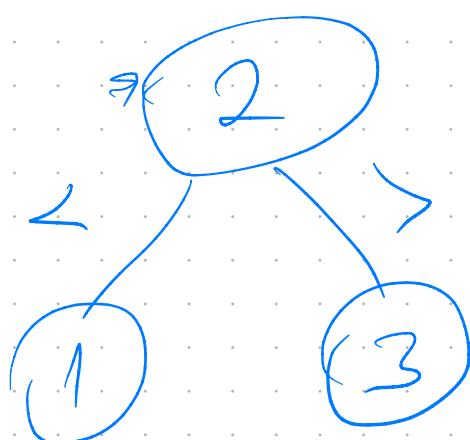
$O(n)$

search  $\rightarrow O(\log n)$

Rebalancing threshold / property

1) Balance

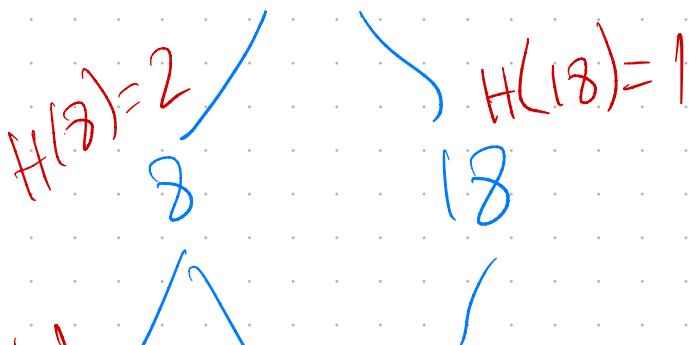
2) Rotations



# 1) Balance

$$H(12) = 3$$

12



Height

$$H(\emptyset) = -1$$

null

node

$$H(\text{single node}) = 0$$

$$H(n) = \max(H(T_L), H(T_R)) + 1$$

$$\begin{aligned} H(4) &= \max(-1, -1) + 1 \\ &= -1 + 1 \end{aligned}$$

$$H(4) = 0$$

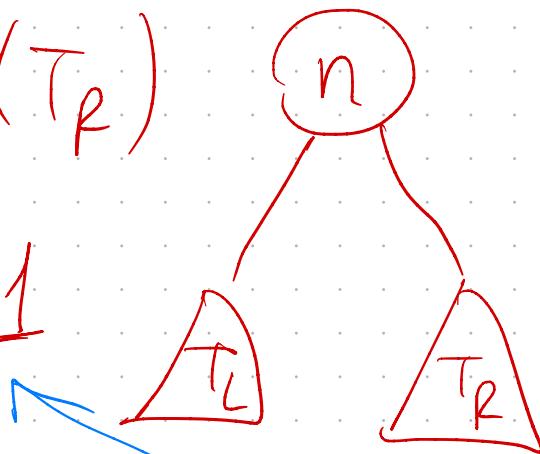
$$\begin{aligned} H(5) &= \max(0, -1) + 1 \\ &= 0 + 1 \end{aligned}$$

$$H(5) = 1$$

## Balance Calculation

$$B(n) = H(T_L) - H(T_R)$$

AVL Tree  $|B(n)| \leq 1$



$$B(1) = 0$$

$$H(1) = 1$$

$$H(3) = 2$$

$$B(3) = 1$$

parameterized

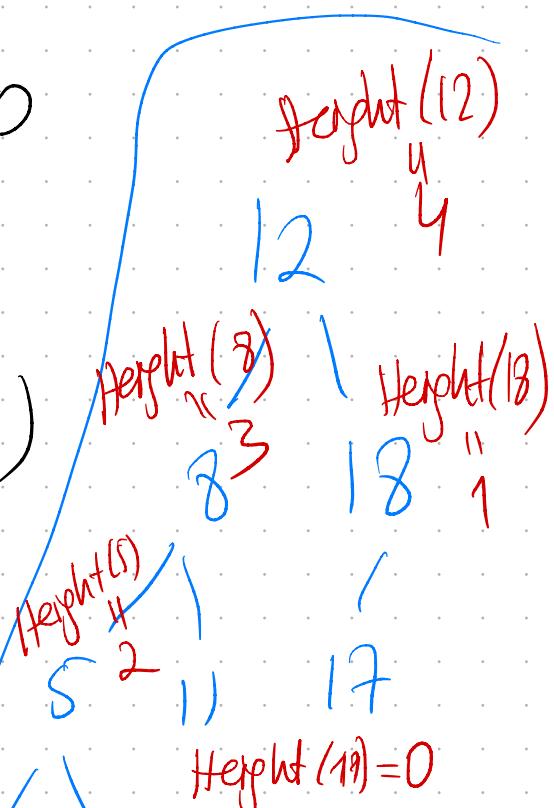
$$H(0) = 0$$

$$B(0) = 0$$

$$H(2) = 0$$

$$\begin{aligned} B(2) &= -1 - (-1) \\ &= -1 + 1 \\ &= 0 \end{aligned}$$

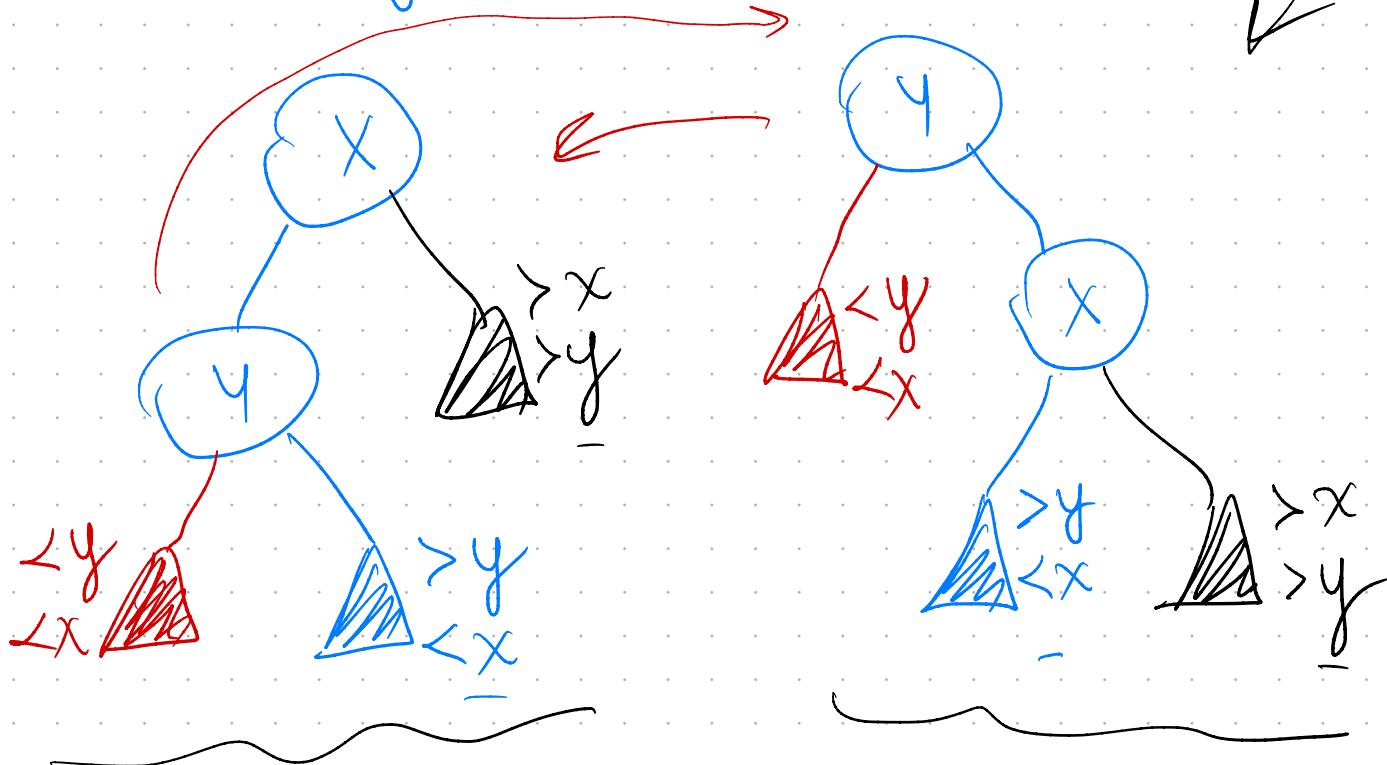
E.g.: Is this an AVL tree?



NO!

## Rotations

Rotations fix imbalance



$x > y$

$x > y$

left-heavy

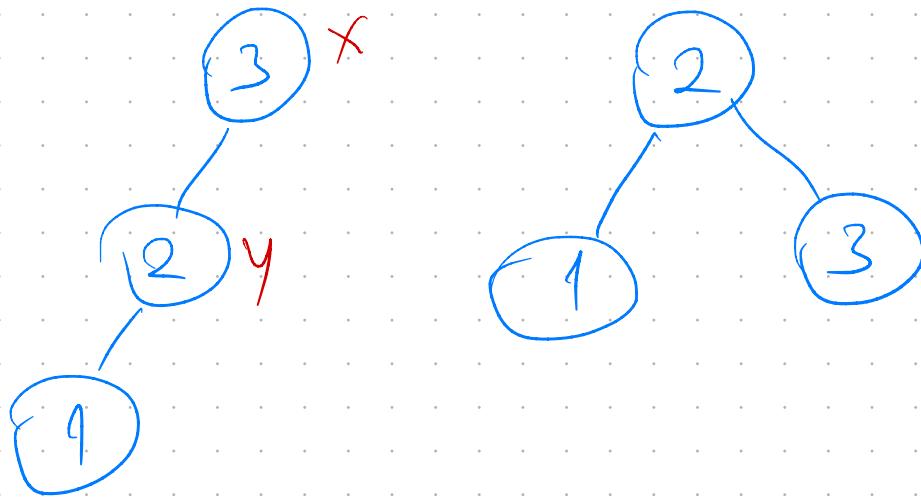
- 1) Right
- 2) Left-right

Right-heavy

- 3) Left
- 4) Right-left

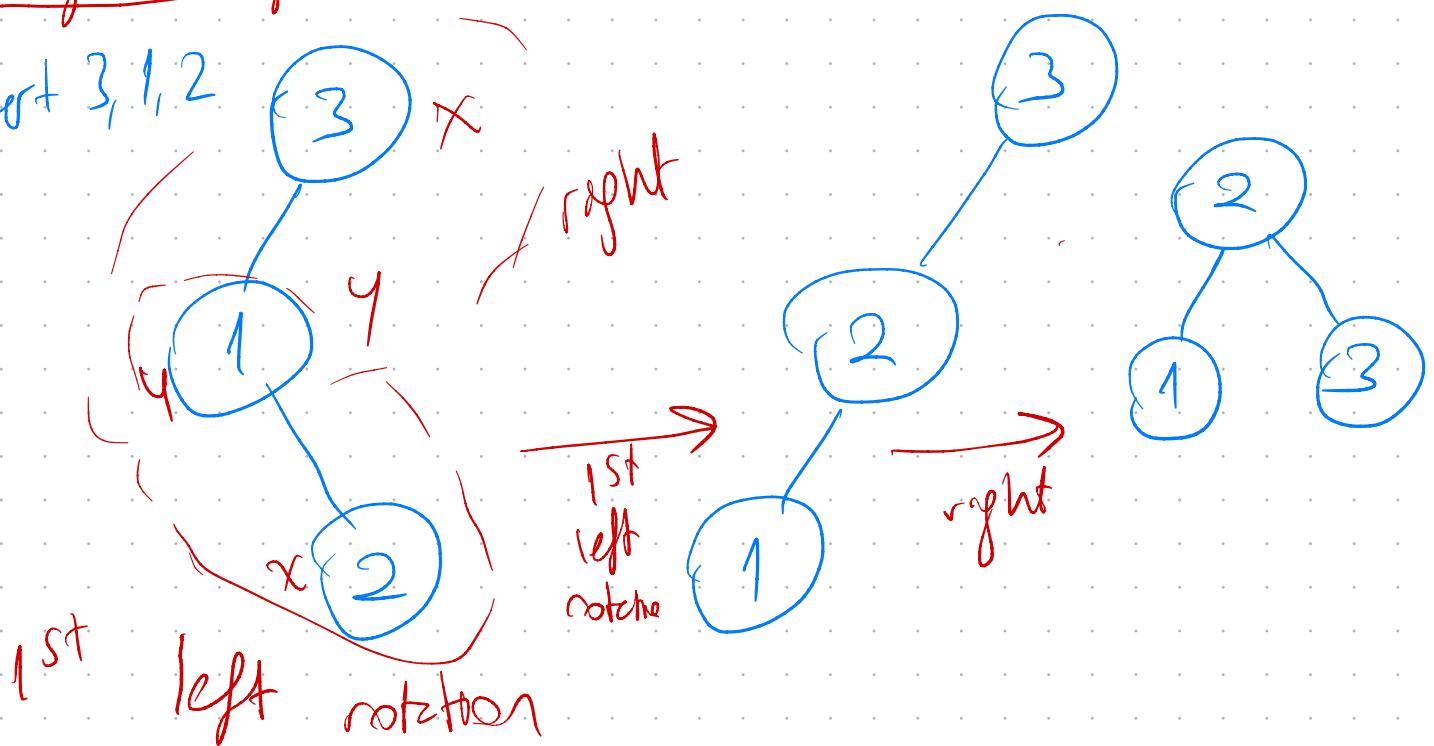
## Right Rotation

Insert 3, 2, 1



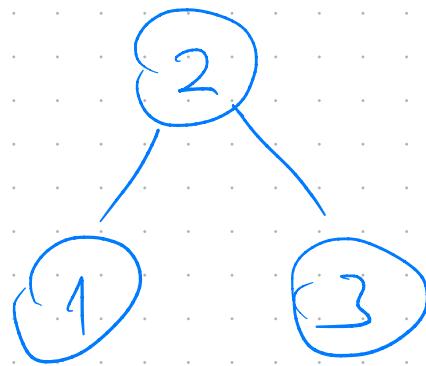
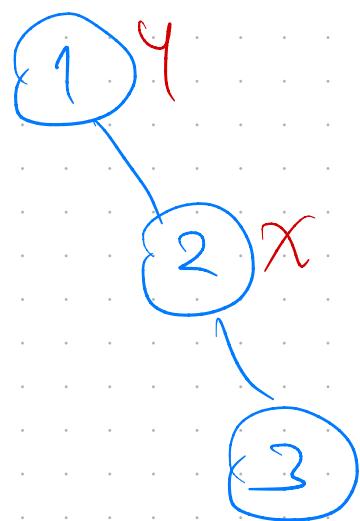
## Left - Right Rotation

Insert 3, 1, 2



## Left Rotation

Insert 1, 2, 3



## Right - Left Rotation

Insert 1, 3, 2

