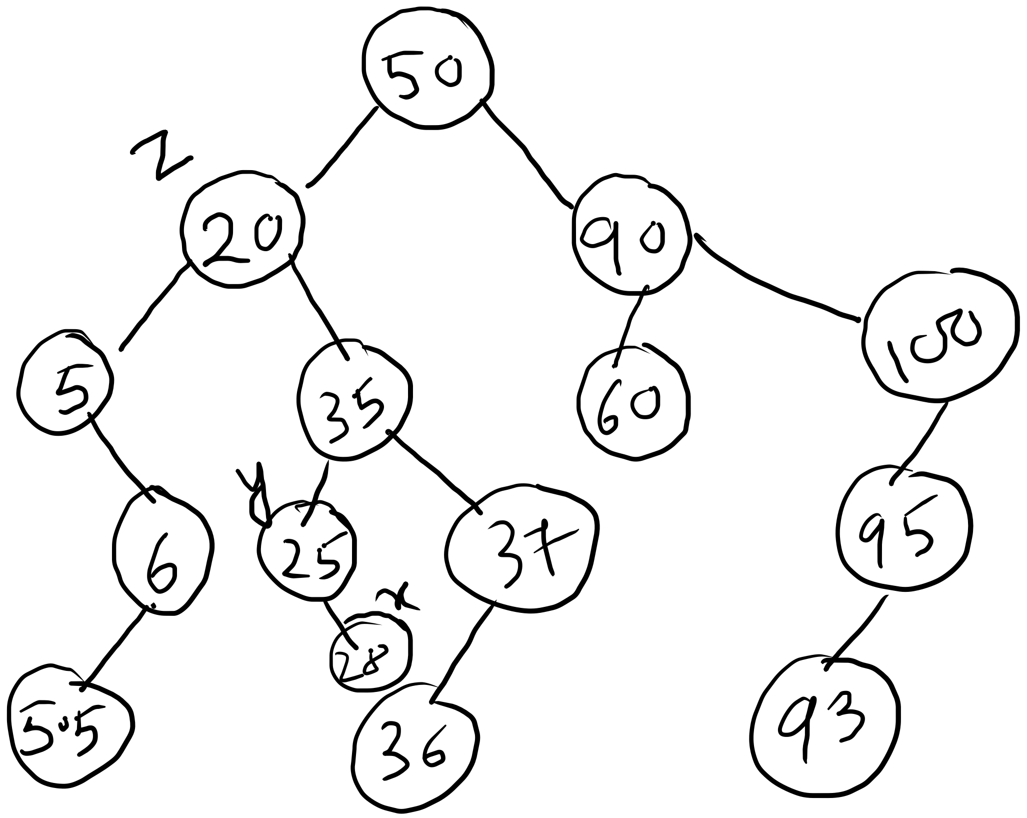
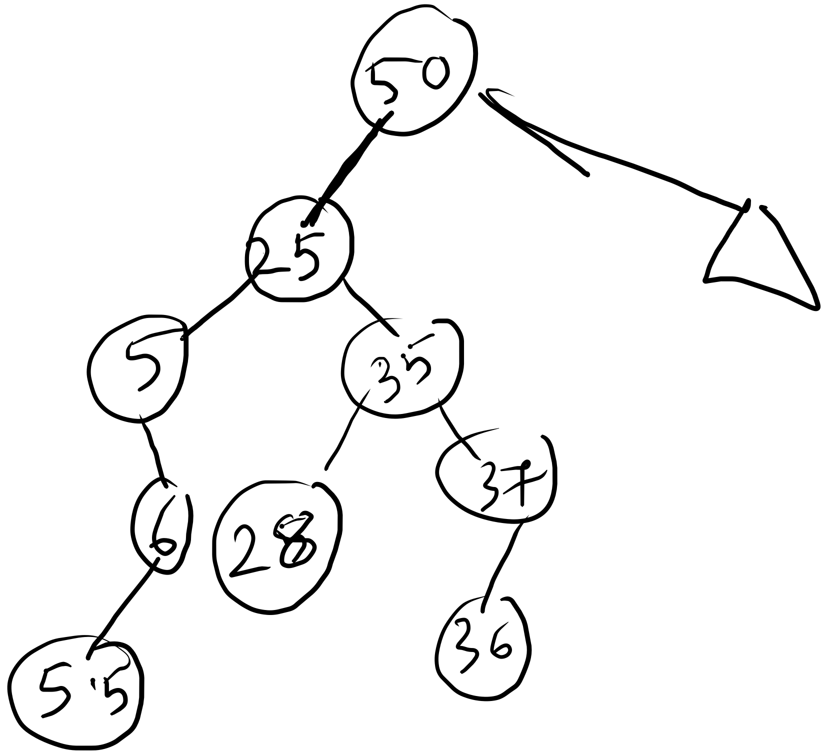
Insert: 50, 20, 90, 5, 35, 6, 37, 60, 5·5, 36, 100, 95, 93, 25, 28,



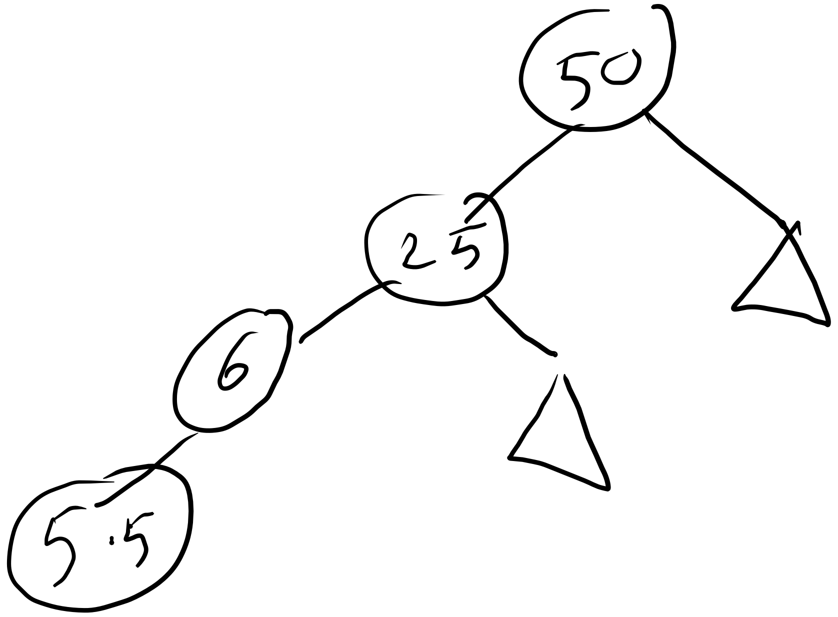
Delete - 20

⇒



Delete 5

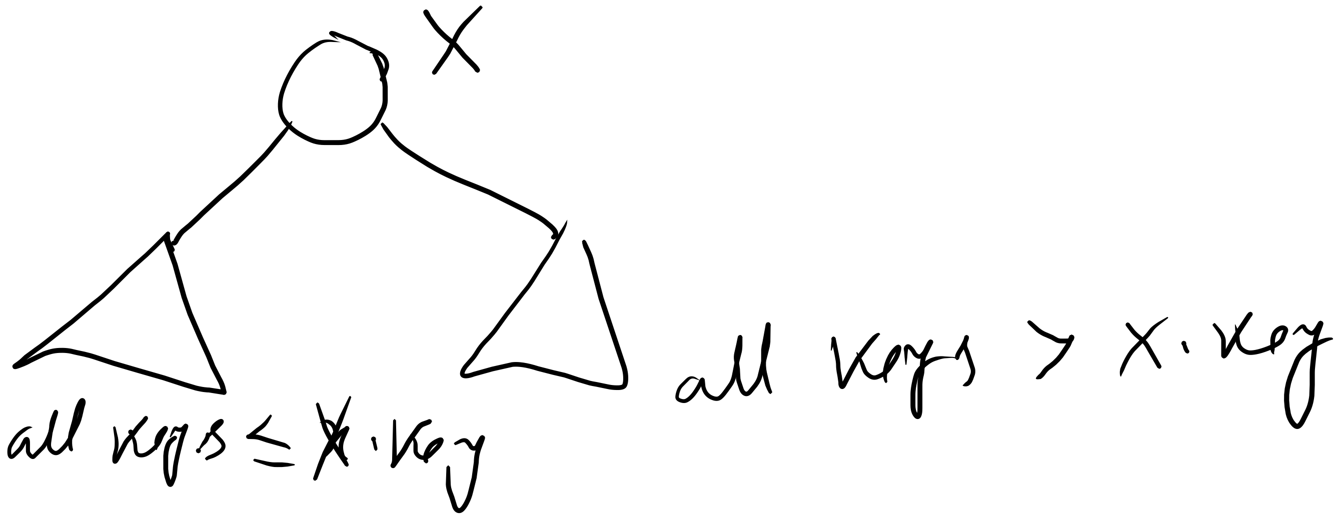
⇐



BST

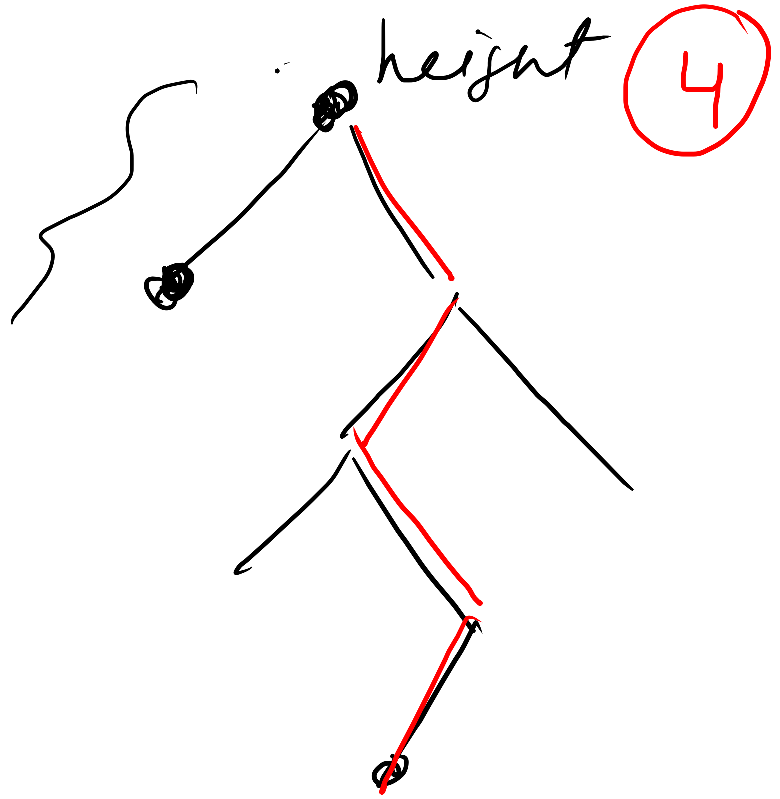
- rooted binary tree
- each node has
  - a key
  - left pointer
  - right pointer
  - parent pointer.

BST property



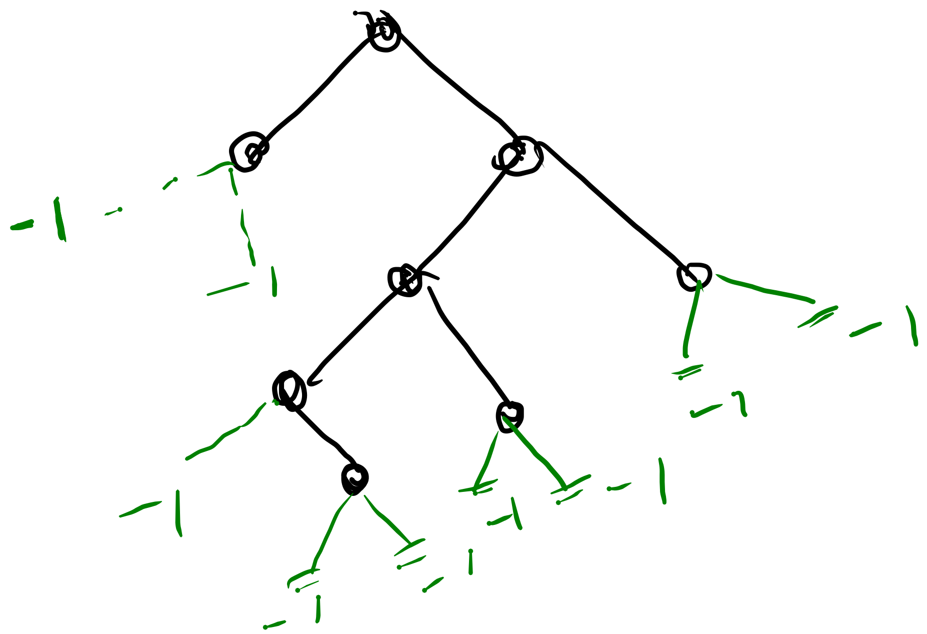
height of a node:

length of the maximum path (# edges) from any leaf node to that node.



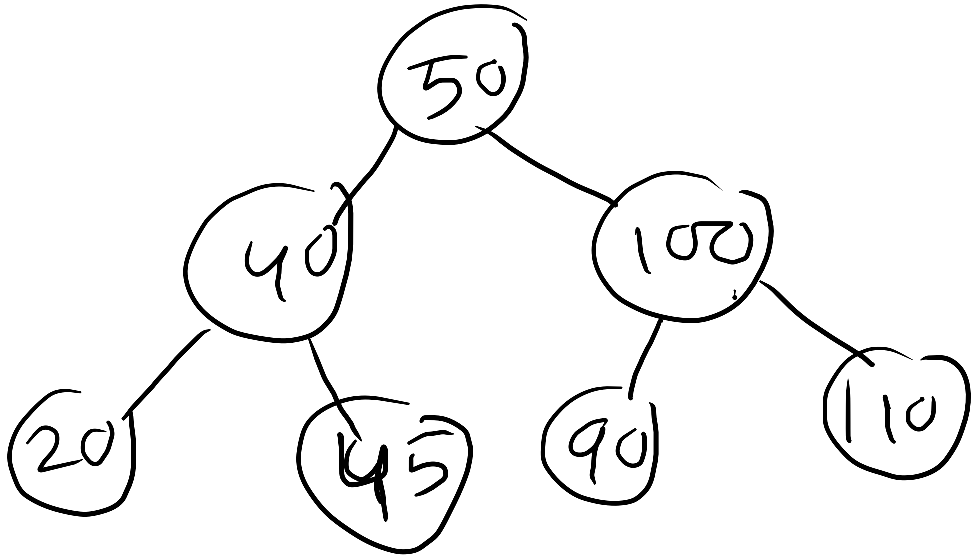
height recursive definition:

height of a node = max { height (left child), height (right child) } + 1



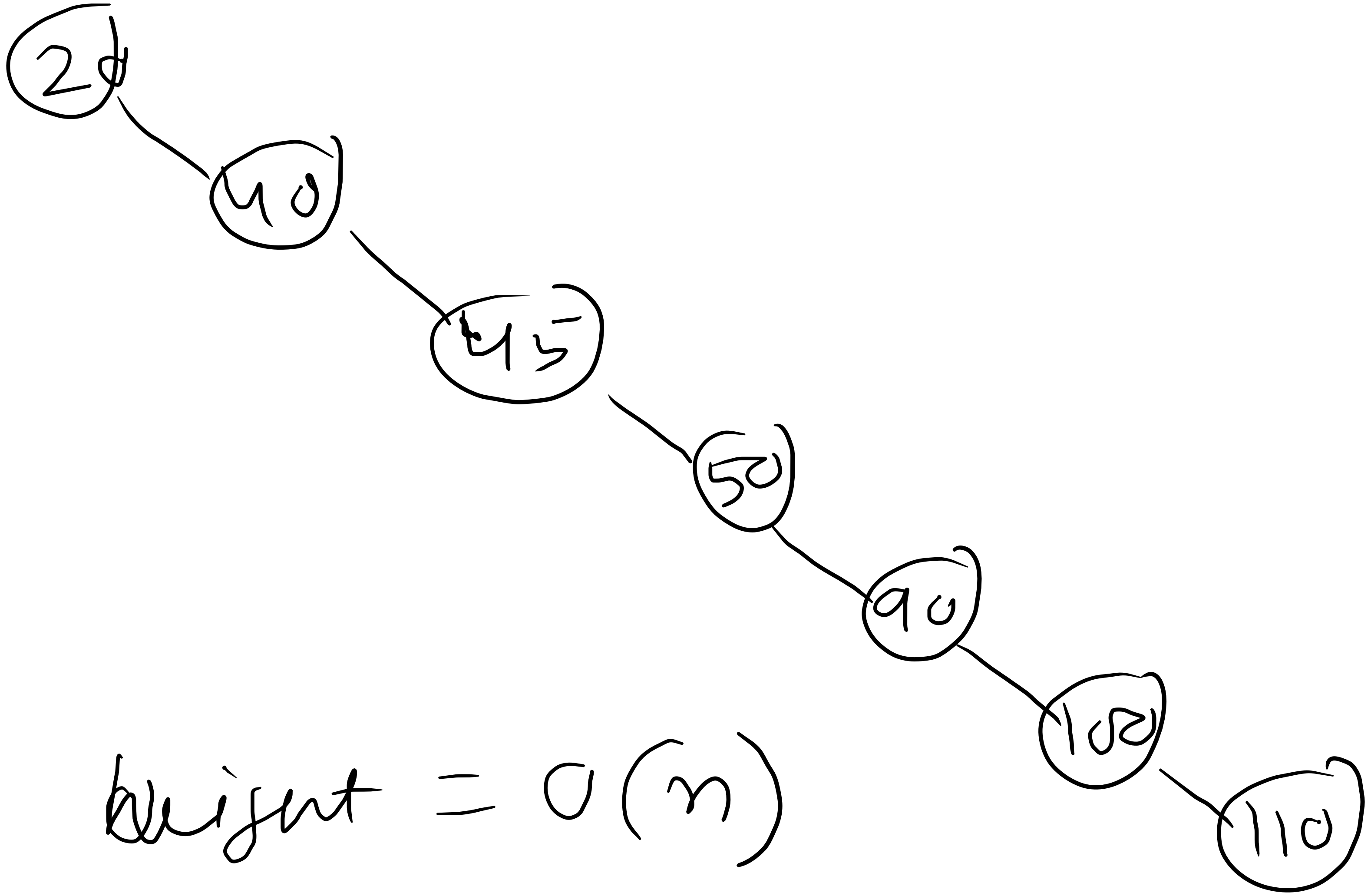
Insert into a BST

50, 100, 90, 110, 40, 20, 45



height = $O(\log n)$

20 40 45 50 90 100 110



height = $O(n)$

All operations except traversal takes $O(\text{height})$ time.

In worst case $O(n)$

can we do better??

Balanced binary Search tree

Different algorithms exist

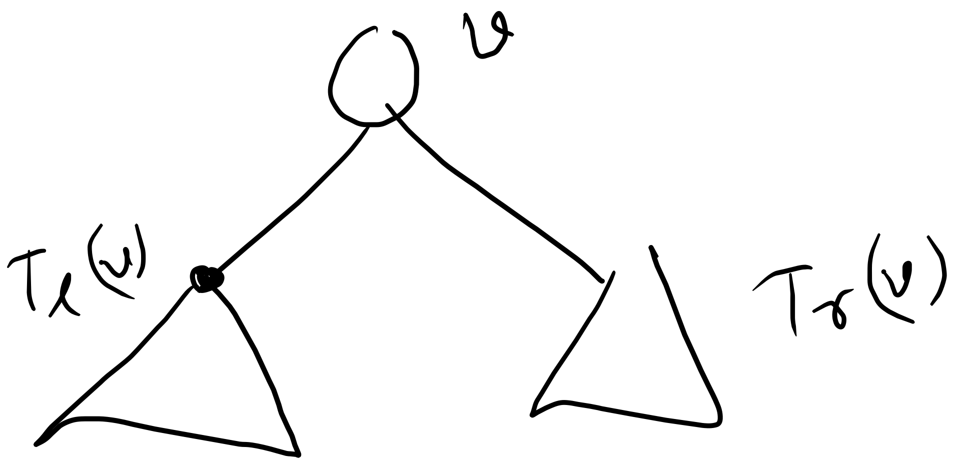
- AVL
- Red-black tree
- Splay tree
- B trees (2-4 trees) multiway search trees.

AVL tree

Balance of a node

It is defined by . bal(v)

$$bal(v) = height(T_\ell(v)) - height(T_r(v))$$

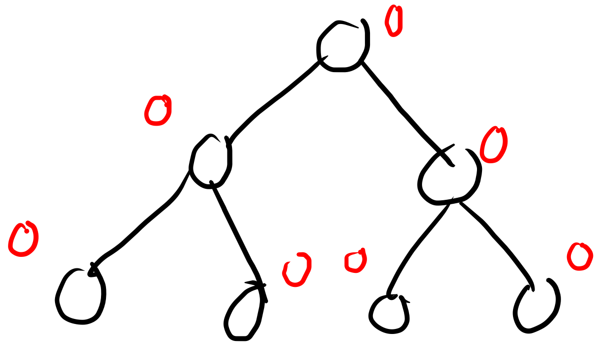


AVL-condition

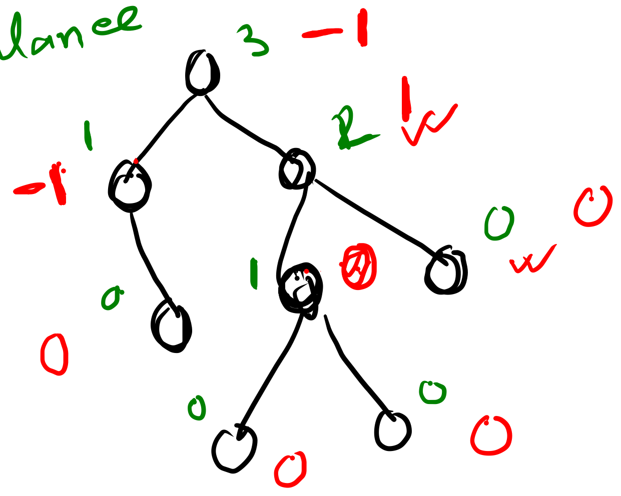
For each node v

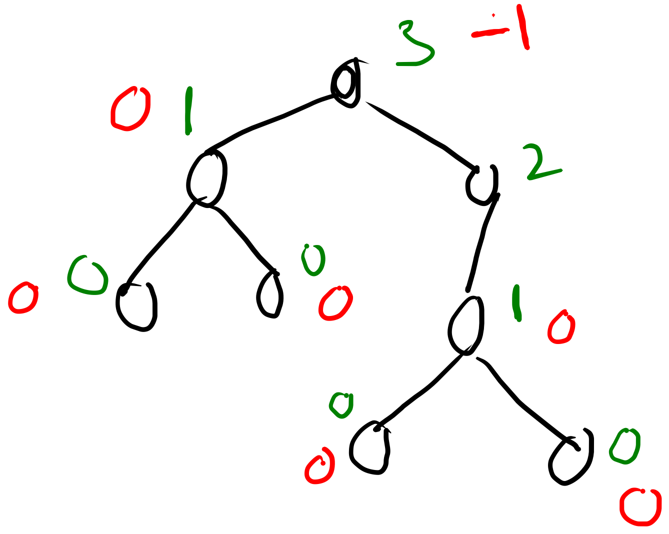
$$bal(v) \in \{0, 1, -1\}$$

Exm



green- height
~~red~~ - balance

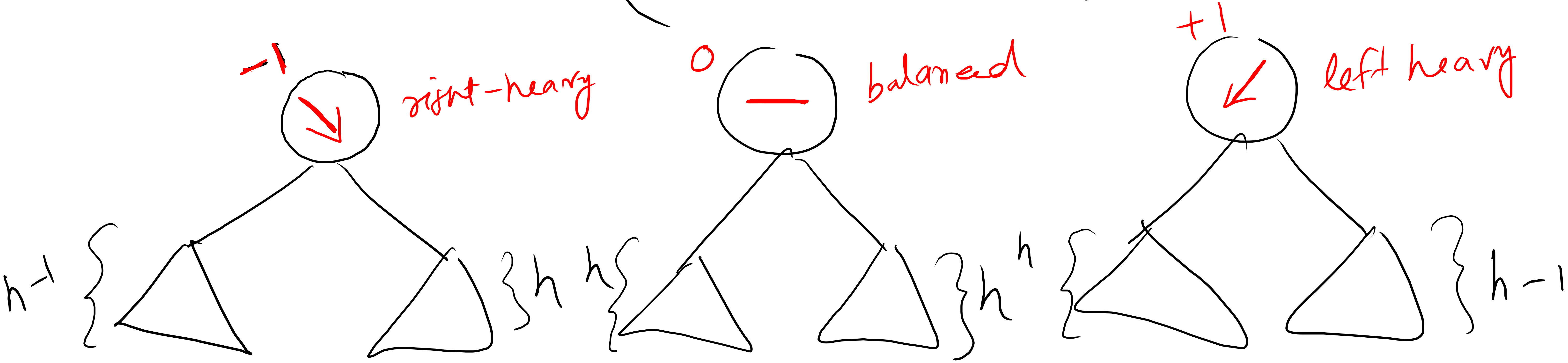




$(1-(-1))=2$

AVL tree:

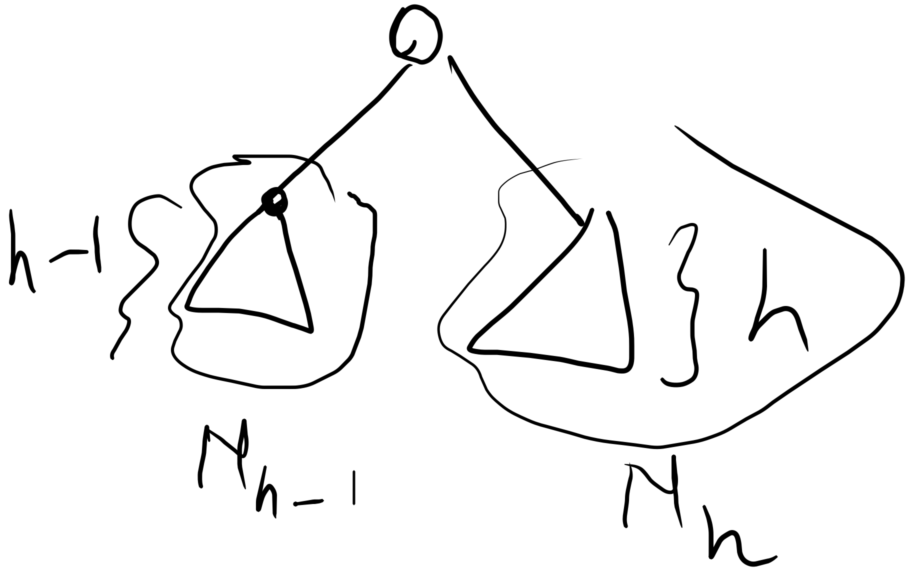
- Every node requires height of left and right children to differ by at most 1
- treat each null tree node as height -1
- Each node stores its height.
  (Datastructure augmentation)



Height of an AVL tree

$N_h \leftarrow$ min # nodes in an AVL tree of height h

$$\boxed{N_h = N_{h-1} + N_{h-2} + 1}$$



$N_h > F_h$

$F_n = F_{h-1} + F_{h-2}$

h-th fibonacci number

$N_h > \frac{\varphi^h}{\sqrt{5}}$

$\Rightarrow \varphi^h < \sqrt{5} \cdot N_h$

$F_h = \frac{\varphi^h}{\sqrt{5}}$ $\varphi = \frac{1+\sqrt{5}}{2} \approx 1.618$

$\Rightarrow h \approx O(\log N_n)$

$\approx O(\log n)$

$\approx 1.440 \lg n$

$$N_n = \underline{\underline{N_{n-1}}} + \underline{\underline{N_{n-2}}} + 1$$

$$N_{n-1} > N_{n-2}$$

$$N_n > N_{n-2} + N_{n-2} + 1 \Rightarrow N_n > 2\,N_{n-2} + 1$$

$$> 2 \cdot 2 \cdot N_{n-4} + 1$$

$$\vdots$$

$$> 2^{n/2} \cdot N_0$$

$$N_n < N_{n-1} + N_{n-1} + 1$$

$$N_{n-2} > N_{(n-2)-2} + 1 = N_{n-4}$$

$$N_{n-4} > N_{n-4)-2} = N_{n-6}$$

$$2^{n/2} N_0 < N_n$$

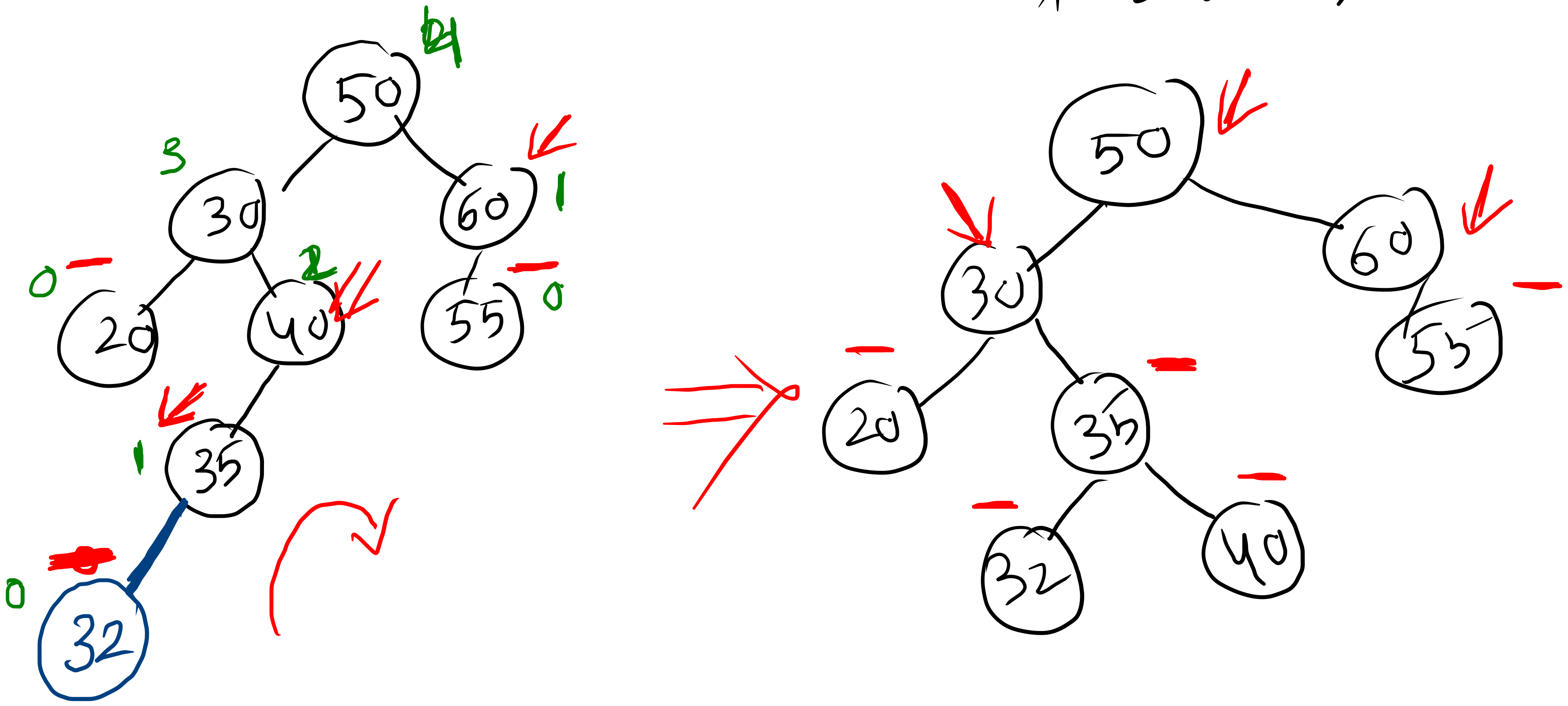
$$2^{n/2} < N_n$$

$$n = O(\log N_n)$$

AVL tree insertion

50, 30, 60, 20, 40, 55, 35

Insert → 32



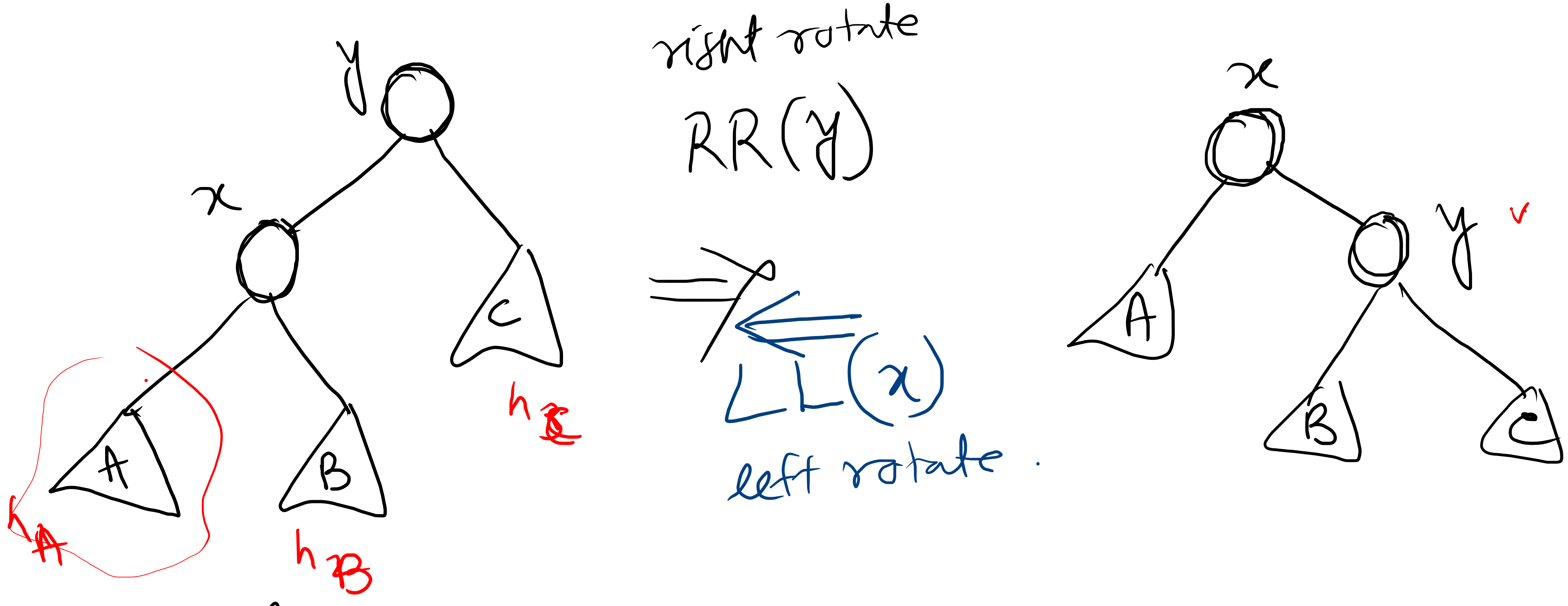
↘

⇘ double right heavy

↙

⇙ double left heavy

—



Inorder:

A x B y C

Inorder

A x B y C