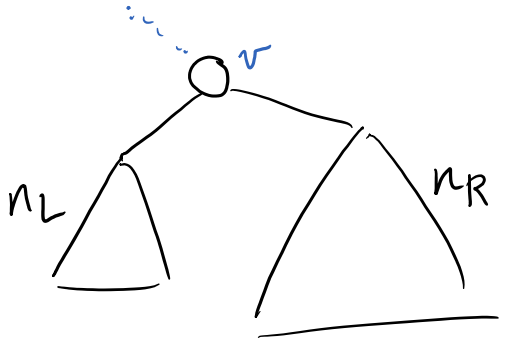
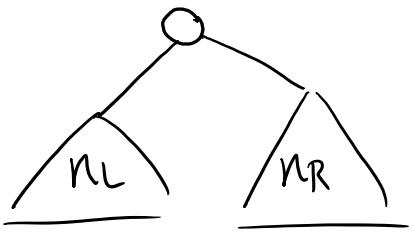
Size-Balanced Property for a node



$$MAX(n_L, n_R) \leq 2 \times MIN(n_L, n_R) + 1$$

A tree is size-balanced iff all nodes are size-balanced. [NOT JUST GLOBAL ROOT].

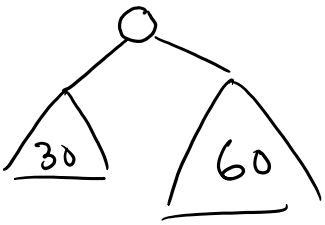
EX SUPPOSE $N = 91$



$n_L + n_R = 90$

$60 \stackrel{?}{\leq} 2 \times 30 + 1$

$60 \leq 61$ ✓



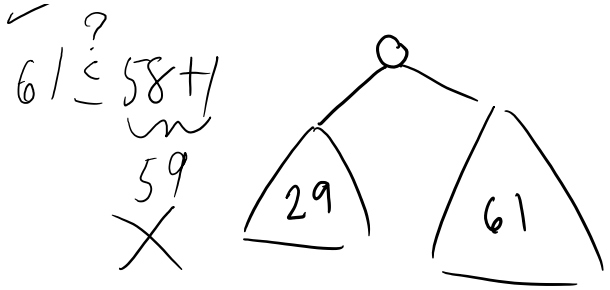
ok?

(a) YES

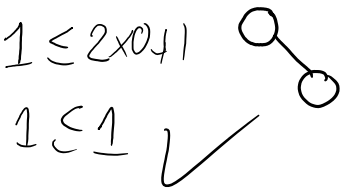
(b) NO

$61 \stackrel{?}{\leq} 58 + 1$

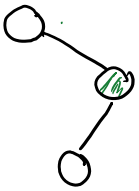
(a) YES



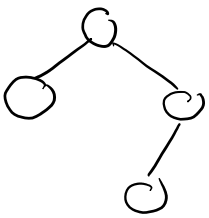
ok? (a) YES
(b) NO



ok? (a) yes
(b) No



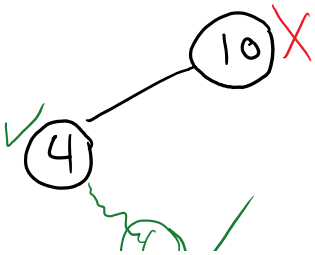
ok? (a) yes
(b) No

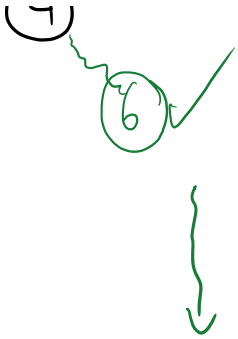


ok? (a) yes
(b) No

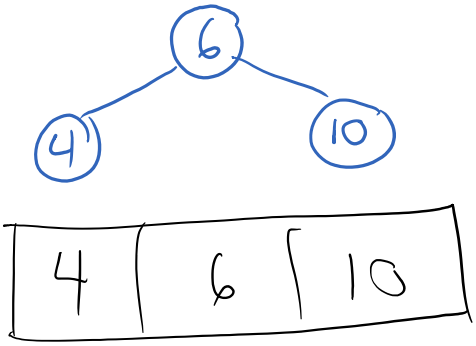
Maintaining size balanced property...

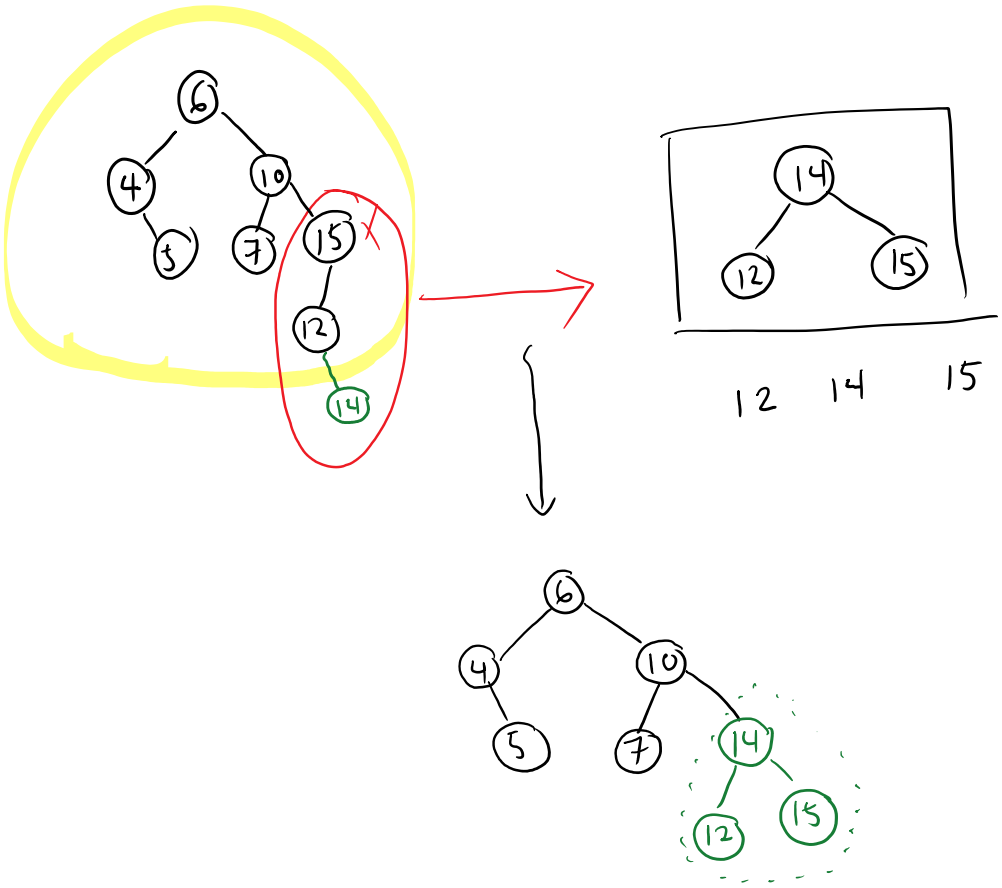
insertion seq: 10 - 4 - 6





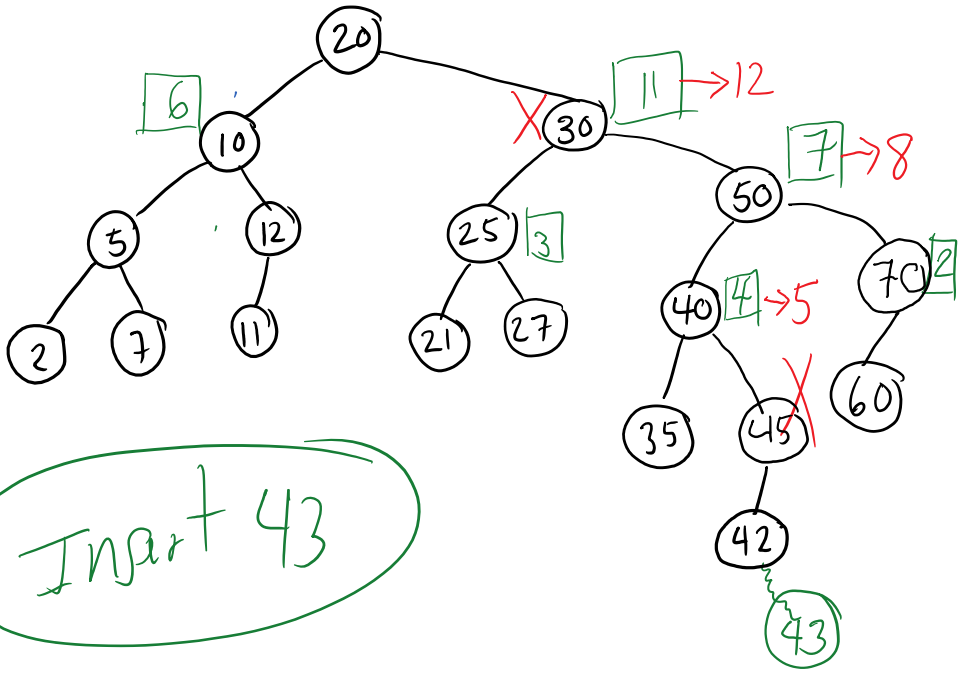
more insertions:



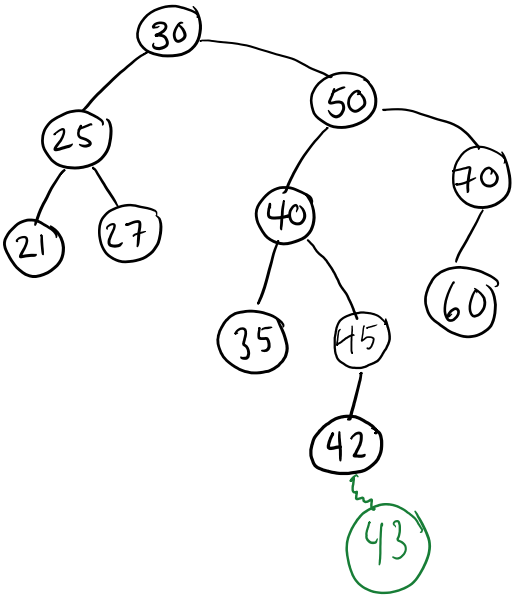


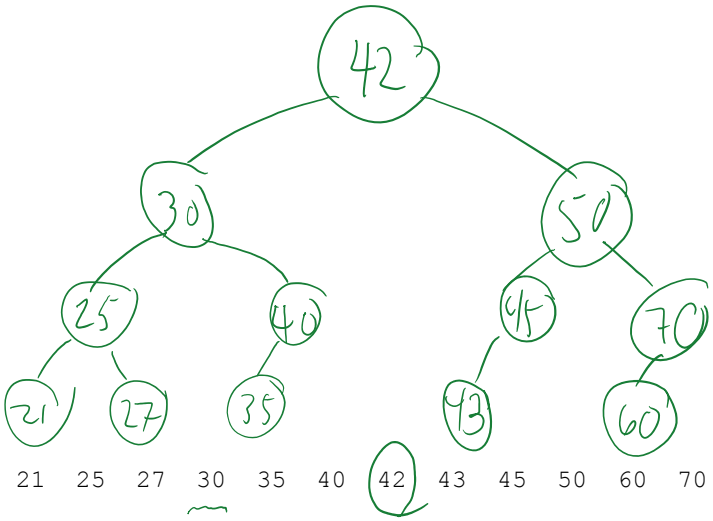
BIGGER EXAMPLE

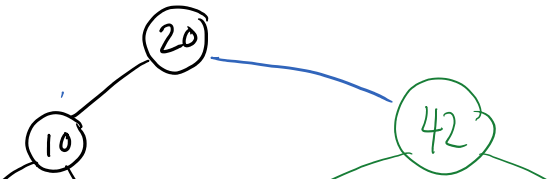
SIZE-BALANCED ?

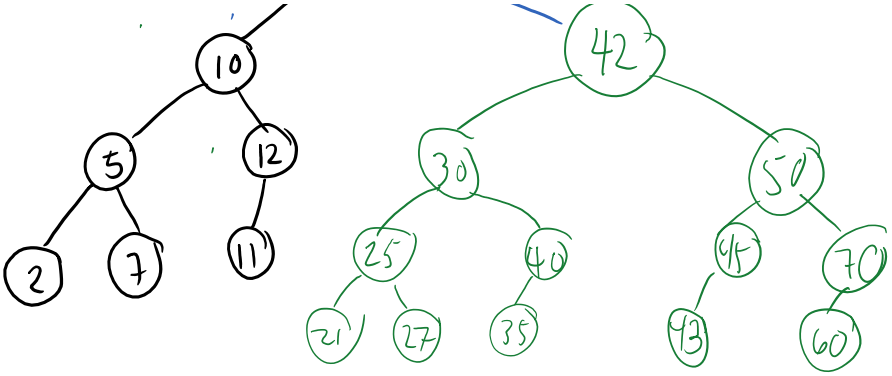


Insert 43

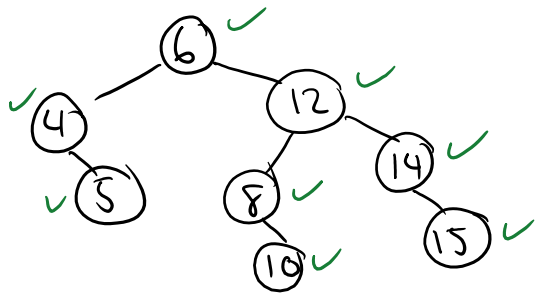








20



MAX HEIGHT OF A SIZE-BALANCED TREE W/ N NODES?

$\approx O(\log N)$

$[\approx \log_{3/2}(n)]$

Intuition of Amortized Runtime Claim...

Claim: a sequence of N insertions into an initially empty Size-Balanced BST takes

O(NlogN) time

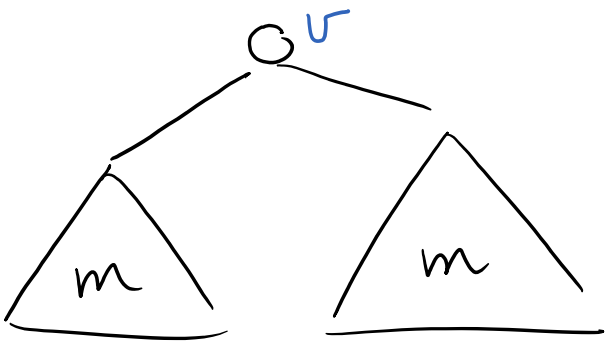
Thus, "on average" each insertion takes O(log N) time.

BUT: Any ***particular*** insertion may **not** be O(log N)

Wort case for one insertion: $\Theta(N)$

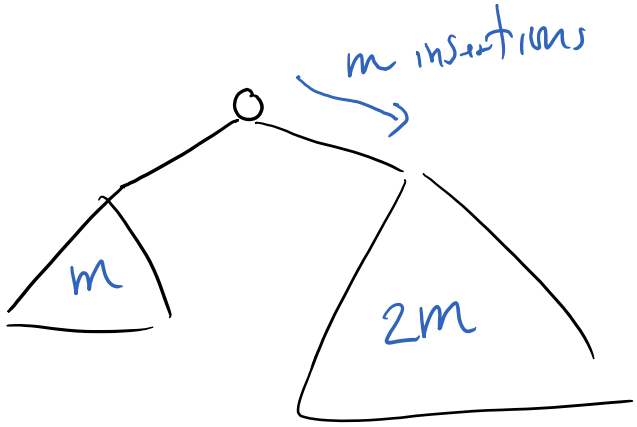
(NOT A PROOF ...)

consider this config:



$N = 2m + 1$

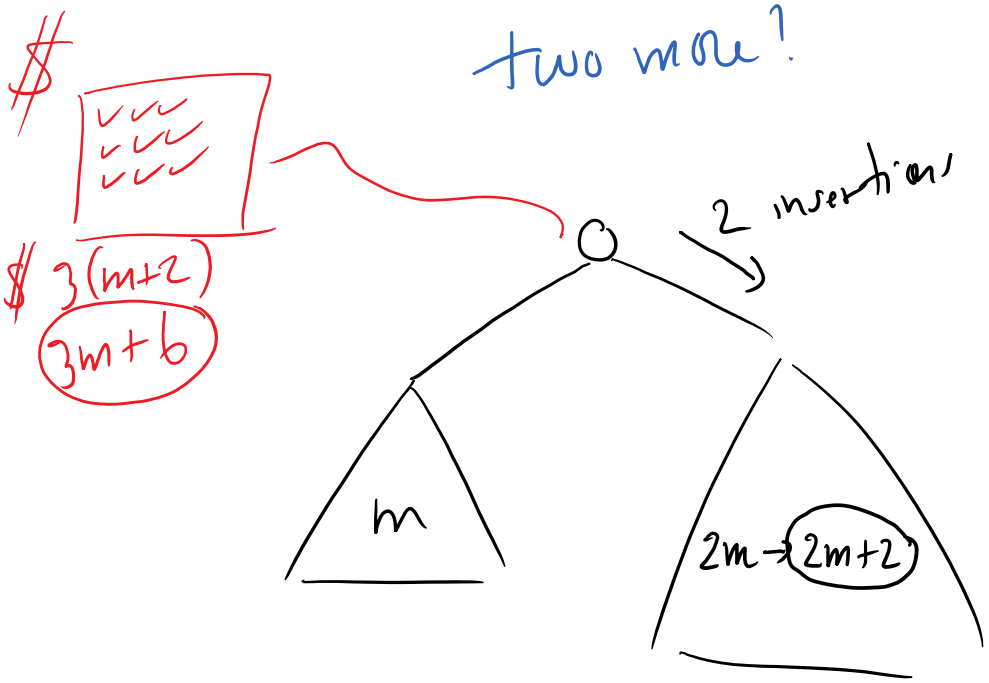
GAME: force (v) to be re-balanced as soon as possible via seq of insertions



m insertions into Right-side...

still ok..

two more!



$N = m+2m+2+1$

$(2m+2) \overset{?}{\leq} 2 \times m + 1$

NOPE!

Rebalance!

$N_{start} = 2m+1$

$N_{final} = 3m+3$

Total Work?

$3m+2$ insertions $O((2m+2)\log(N_f))$

$2m+2$ insertions $\cup$ $(2m+2)\log(N+1)$

$+$ $O(N \log N)$

Rebalance $O(N)$

Total: $N \log N + N = O(N \log N)$