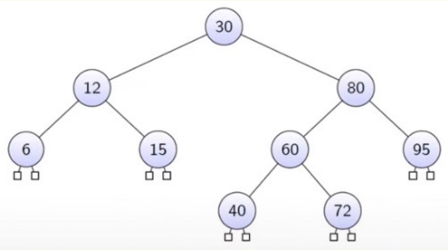
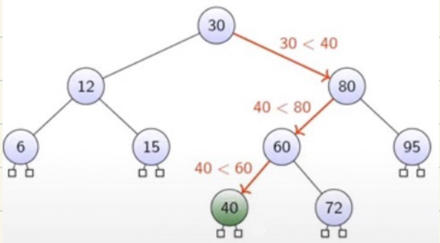
Video 1. Binary Search Tree

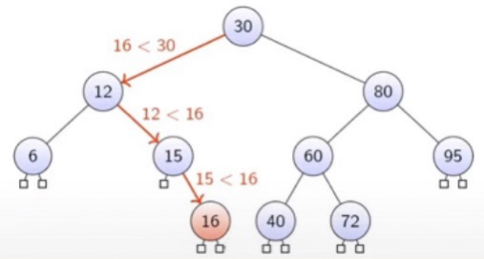
∀node with key k:

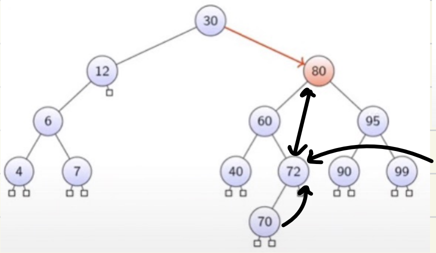
→ keys in left subtree < k

→ keys in right subtree > k









maximal node in left child

swap with node to be removed

recursively remove the node

recurrence equation: $T(0) = c_0$ $T(h) = T(h-1) + c_1$

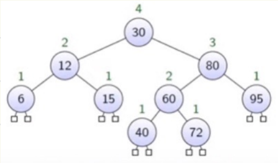
closed form: $T(h) = hc_1 + c_0$ => time complexity $O(h)$

height { best case $O(\log n)$
worst case $O(n)$

Video 2. AVL Tree

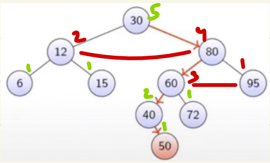
Adelson-Velskii and Landis

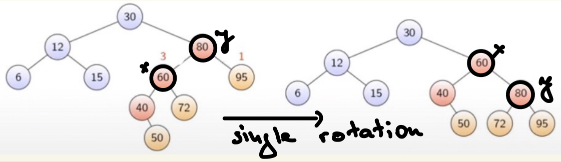
AVL Balance condition: the heights of the children $\Delta h \leq 1$



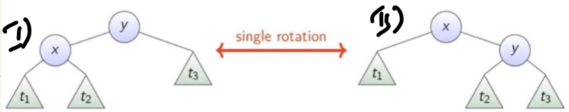
null references as leaves

insert 50 ?





Two-node rotation



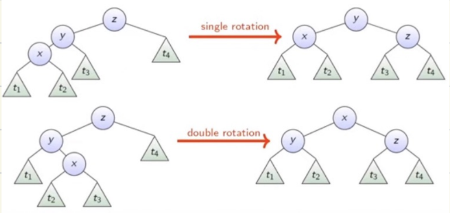
I -> II) $h(x) - h(t_3) > 1$

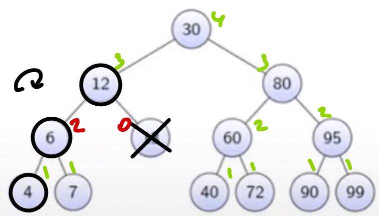
$h(t_1) - h(t_2) > 1$

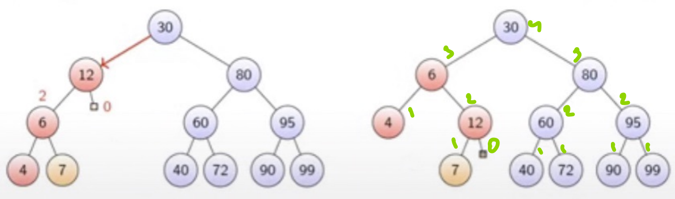
II -> I) $h(y) - h(t_1) > 1$

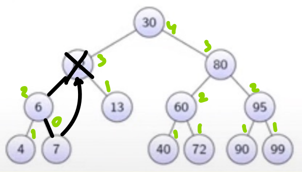
$h(t_3) - h(t_2) > 1$

Tri-node rotations









time-complexity (get, put, remove) = $O(\log n)$

tri-node restructuring = $O(1)$

get, put, remove -> $O(h)$ nodes + $O(h)$ restruc.

$h = O(\log n)$

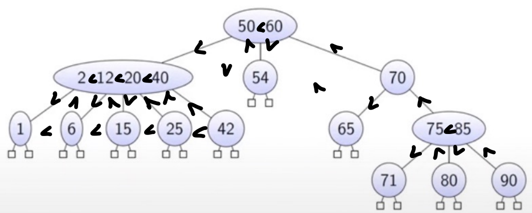
Video 3. (2,4) Tree

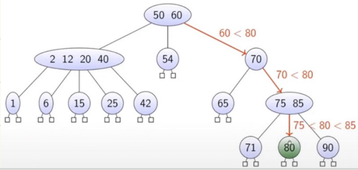
internal node → at least 2 children

d children → ordered list of $k_1, \dots, k_{d-1}$ keys

$\forall k$ in i-th child → $k_{i-1} < k < k_i$ $k_0 = -\infty$ $k_d = \infty$

} multiway search tree

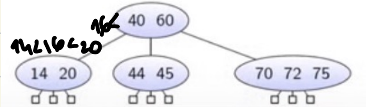




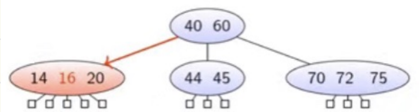
best case: O(log n)
worst case: O(n)

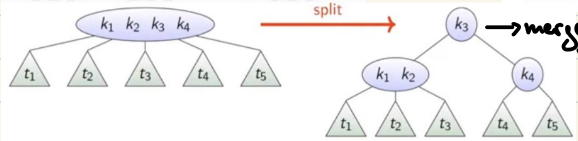
size: at most 4 children

depth: external nodes -> same depth

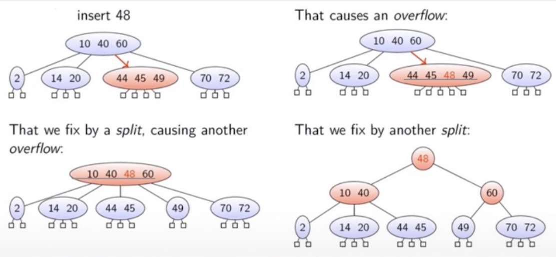


insert 16





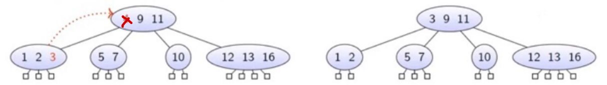
-> merge with parent

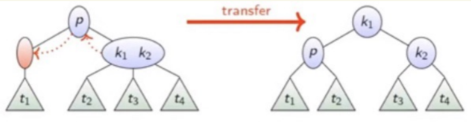

insert 48

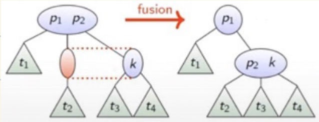
That causes an *overflow*:

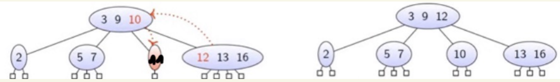
That we fix by a *split*, causing another *overflow*:

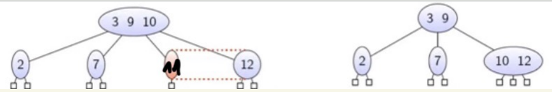
That we fix by another *split*:









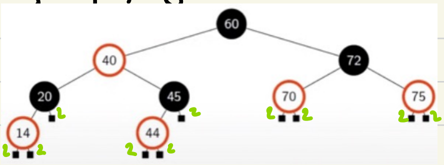


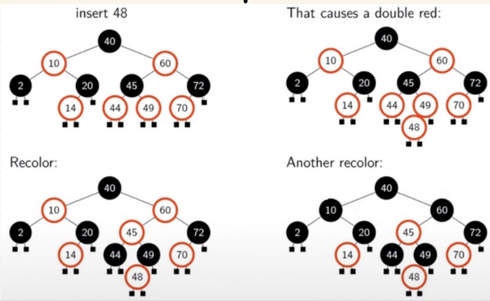
get, put, remove = O(log n)
h = O(log n)
split, transfer, fushion = O(1)
get, put, remove -> O(h) nodes

Video 4.

Red-Black Tree

root property: the root is black
external property: every leaf is black
red property: the children of a red node are black
depth property: all leaves have the same black depth

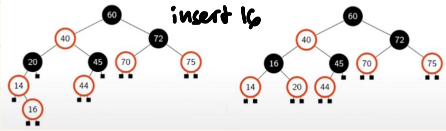


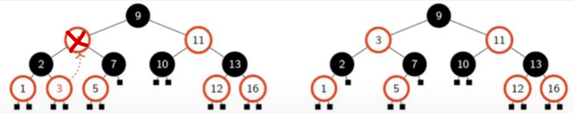

insert 48

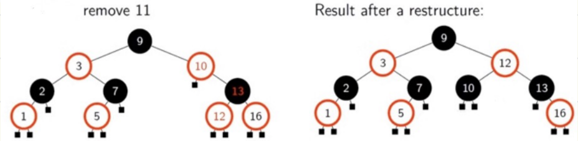
That causes a double red:

Recolor:

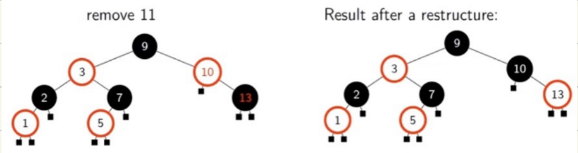
Another recolor:


insert 16




remove 11

Result after a restructure:


remove 11

Result after a restructure:

get, put, remove = O(log n)
height = O(log n)
recoloring = O(1)
get, put, remove -> O(h) nodes