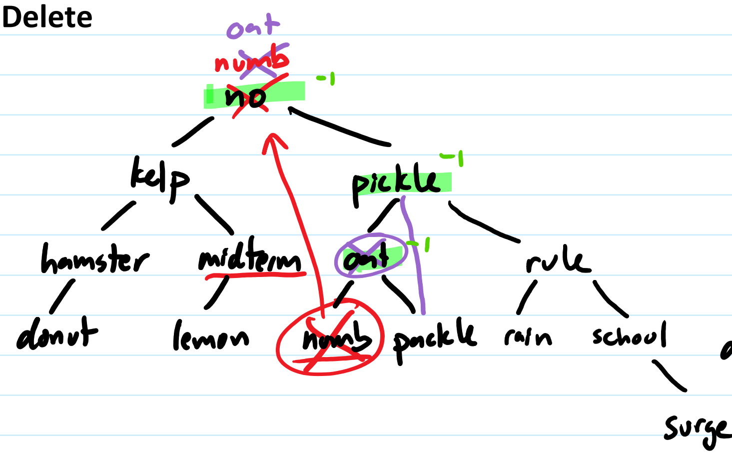
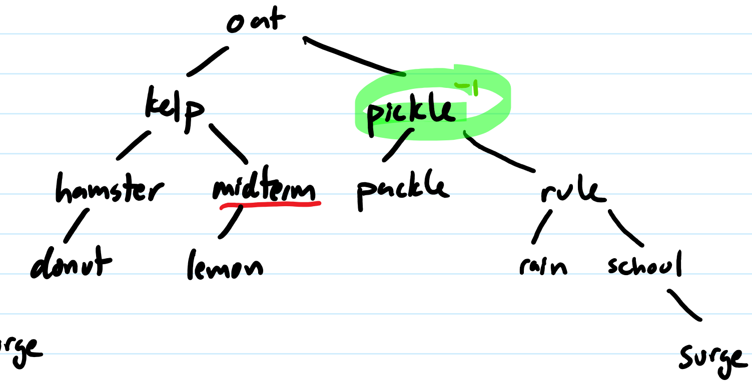
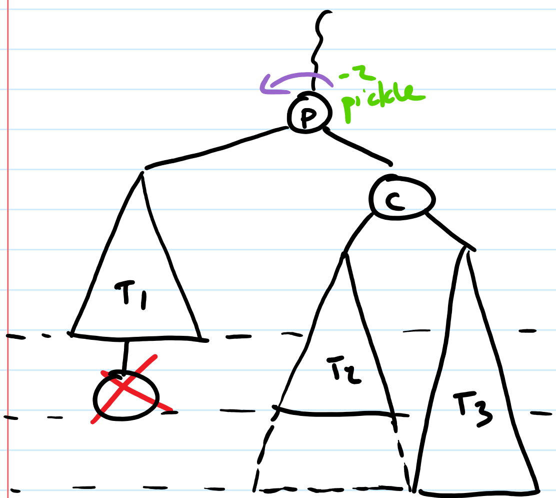
AVL Delete



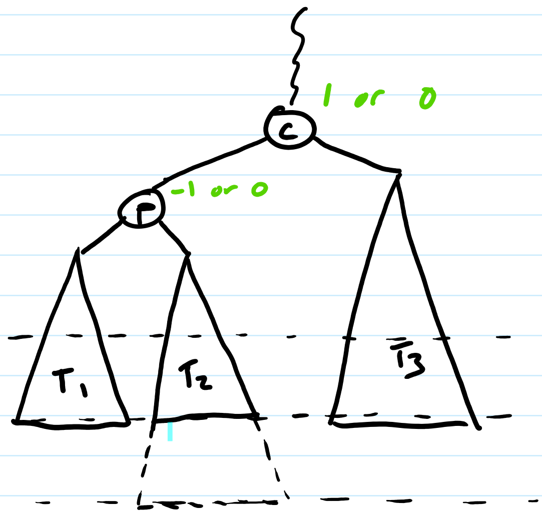
oat
numb
no
-1
kelp
pickle
-1
hamster
midterm
oat
-1
rule
donut
lemon
numb
pickle
rain
school
surge



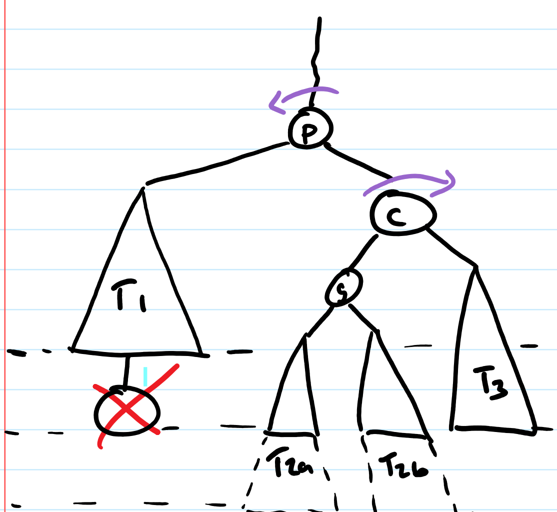
oat
kelp
pickle
-1
hamster
midterm
pickle
rule
donut
lemon
rain
school
surge



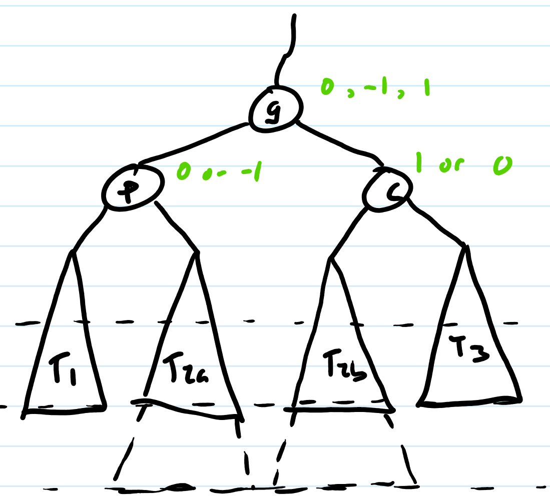
-2
pickle
p
c
T1
T2
T3



1 or 0
c
-1 or 0
p
T1
T2
T3



p
c
g
T1
T3
T2a
T2b

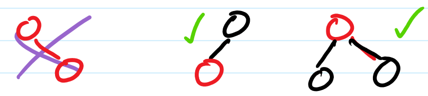


0, -1, 1
g
0 or -1
p
1 or 0
c
T1
T2a
T2b
T3

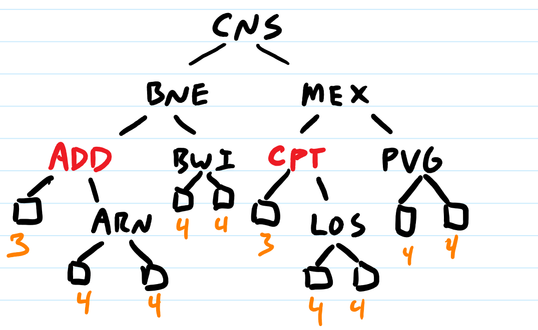
Red-Black Trees

Binary Search Trees such that

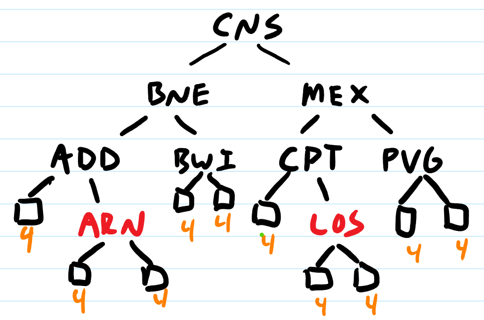
1) each node has a color : red or black
2) root is colored black
3) all leaves are colored black
   ↳ no children
4) children of red nodes are black
5) every path root → leaf has same number of black nodes



add dummy nodes s.t. all data nodes have 2 children; dummy nodes always black



not a red-black tree (violates #5)



red-black tree !

all nodes have 0 or 2 children

non-leaf

leaves in a non-empty full binary tree = # internal nodes + 1

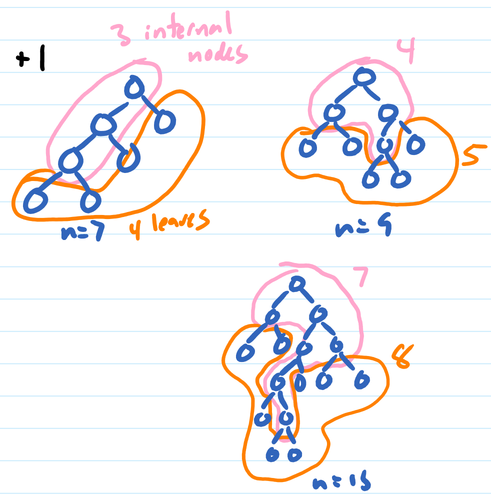
data nodes

if height $O(\log_2 \text{total nodes})$

$= O(\log_2 2 \cdot \#\text{data nodes} + 1)$

$= O(\log 2n+1)$

$= O(\log n)$

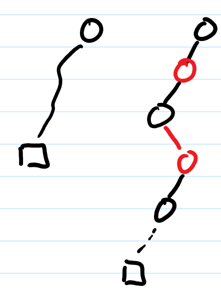


(black height)

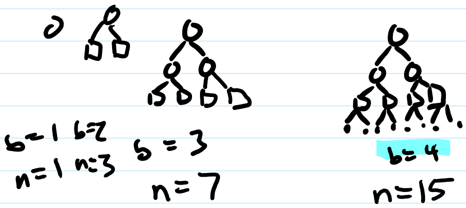
suppose # black nodes in path root → leaf is b

shortest possible path has b nodes

longest possible path has 2b−1 nodes



Red-black tree with black height b has $\geq 2^b - 1$ nodes



$n \geq 2^b - 1$

$n+1 \geq 2^b$

$\log_2 n+1 \geq b$

$b \leq \log_2 n+1$

$h \leq 2b-1 \leq 2\log_2 n+1$

h is $O(\log_2 n)$

$O(\log n)$

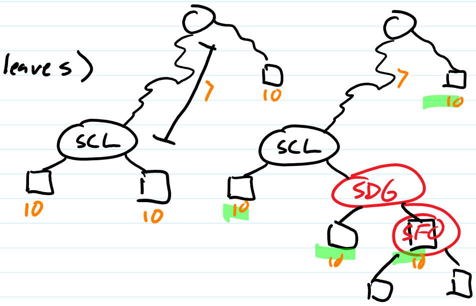
1) Do normal BST insert, color new node red (w/ black dummy nodes as leaves)

2) If parent of new node exists and is black, DONE!

Else

let z = new node

$O(\log n)$ iterations → while z is not root and z -> parent -> color = red

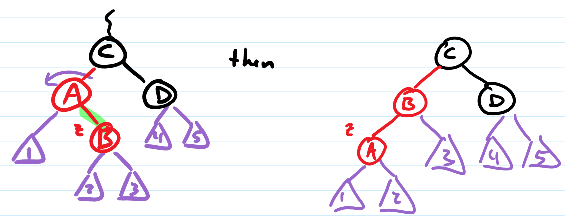


if parent's sibling is also red

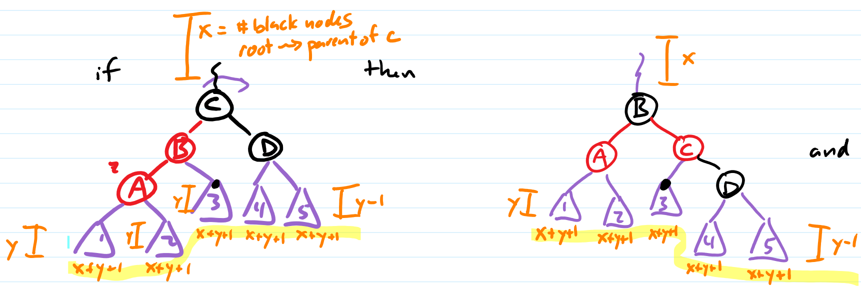
else

$O(1)$ per iteration

if



if



and DONE

[other cases symmetric]

t → root → color = black

Example

ARN CNS CPT BWI MEX ADD PVG LOS BNE EZE