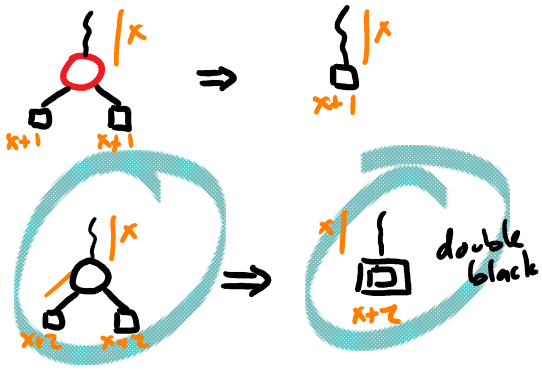
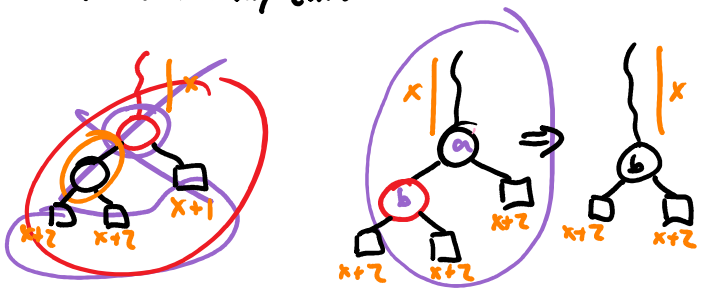
Red-Black Tree Delete

Do normal BST delete; if deleted node has 2 non-leaf children then moved node takes deleted's color
do 1 non-leaf delete where moved node was

0 non-dummy children



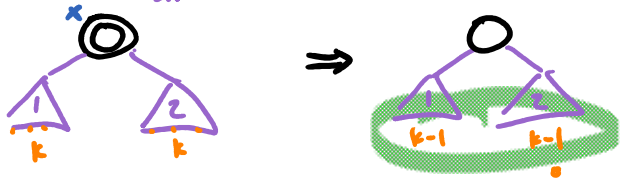
1 non-dummy child



while x is doubly black

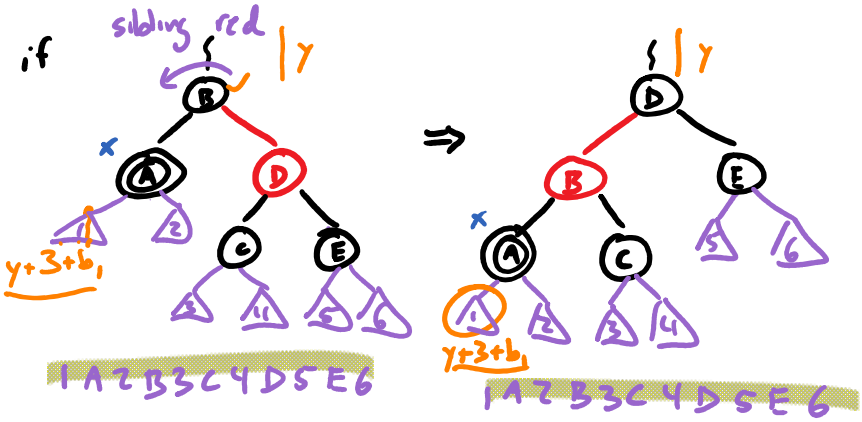
x (double-black node) at root

if



DONE

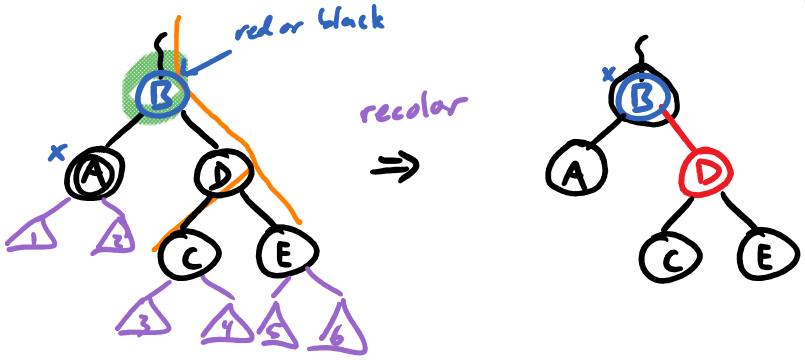
if



1A2B3C4D5E6

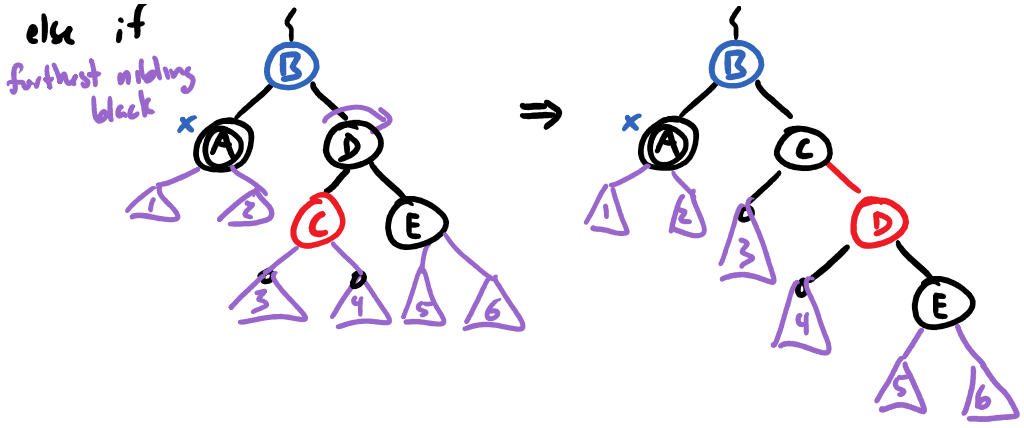
1A2B3C4D5E6

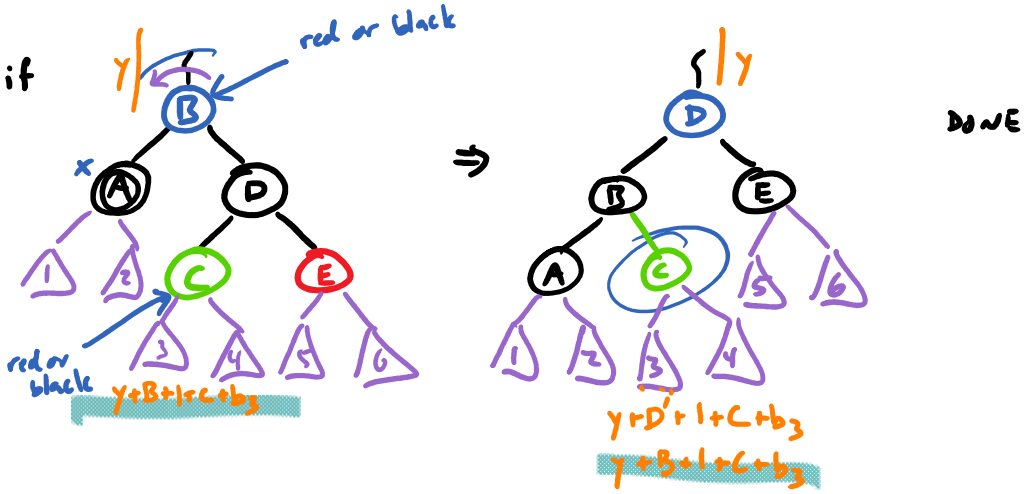
if siblings both black



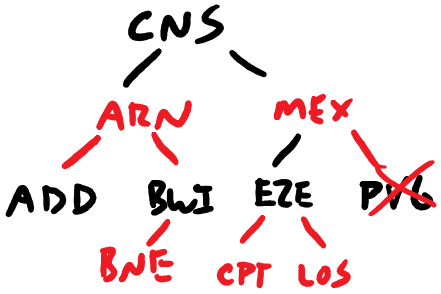
parent black A→B	old color of B	new color of B
○ + ◯	=	○
○ + ○	=	◎

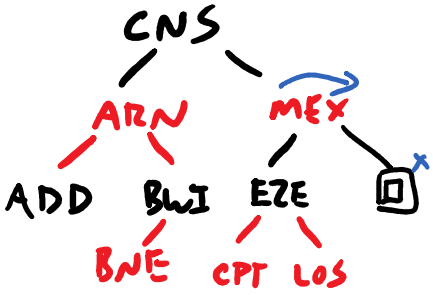
else if furthest nibling

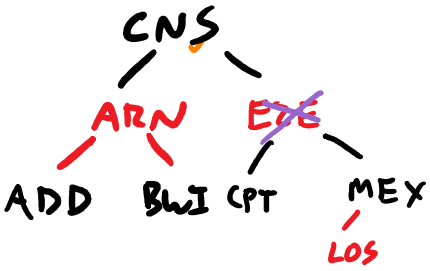

else if
furthest nibling
black
x
B
A
D
C
E
1
2
3
4
5
6


if
y
red or black
B
x
A
D
C
E
red or
black
y+B+1+C+b3
y+D'+1+C+b3
y+B+1+C+b3
DONE

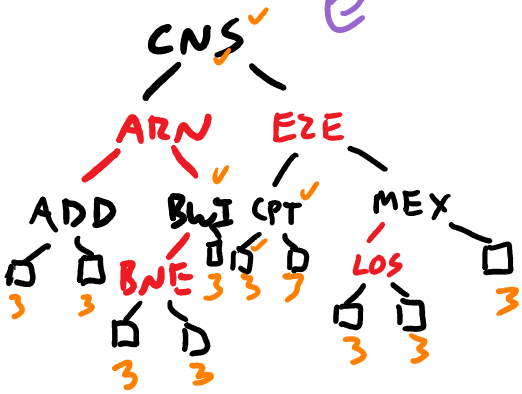
Example Red-Black Delete

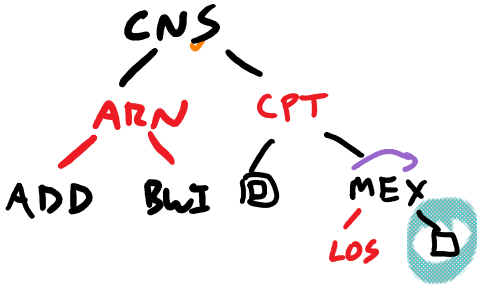


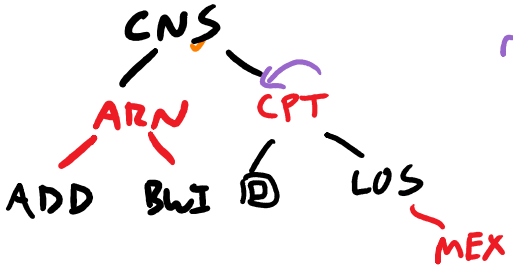




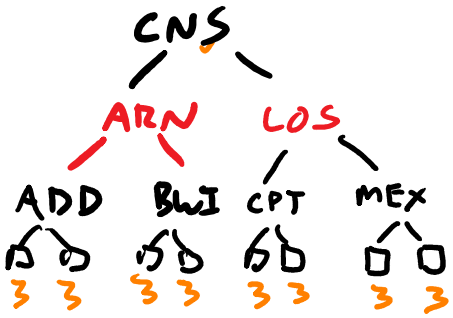
rotate/recolor





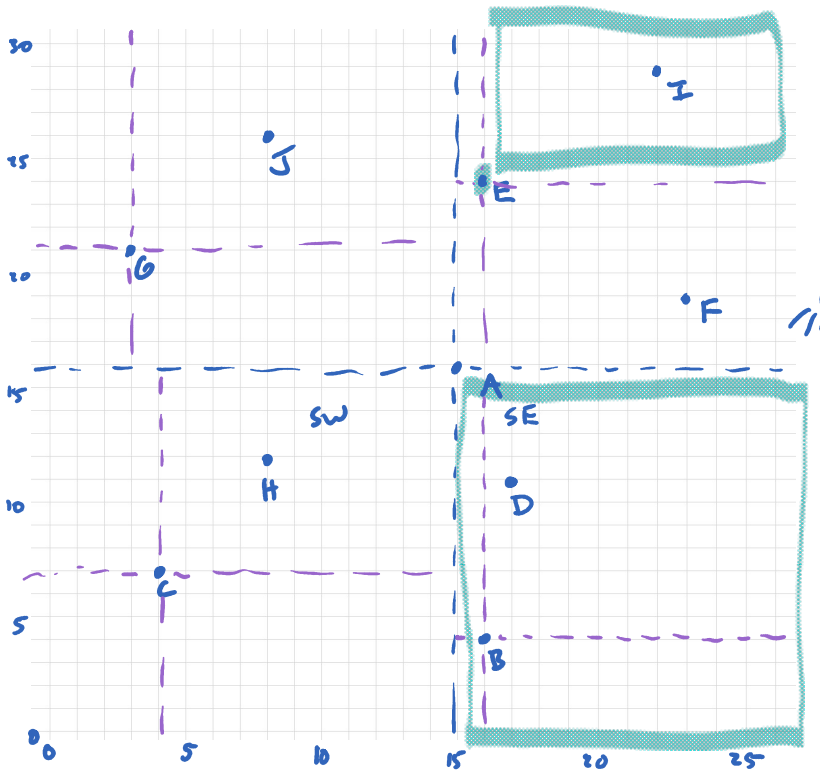


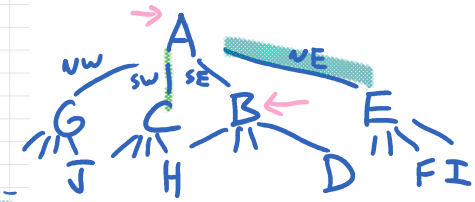
rotate/recolor



Quadtrees

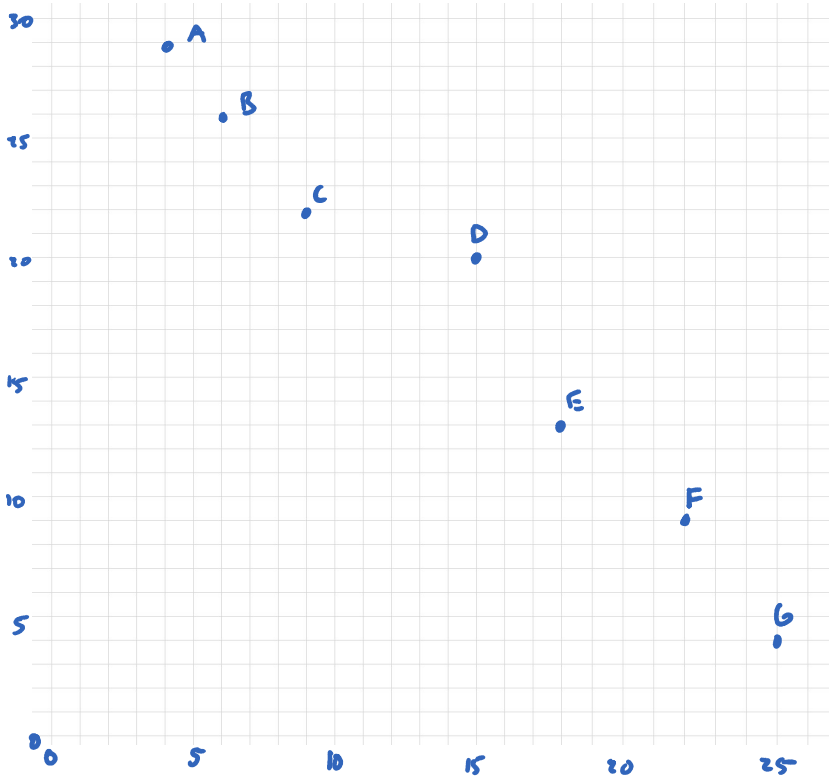
properties: everything in NW subtree of node n is to NW of pt in n
SW " " n " SW of pt in n
SE " " n " SE " "
NE " " n " NE " "

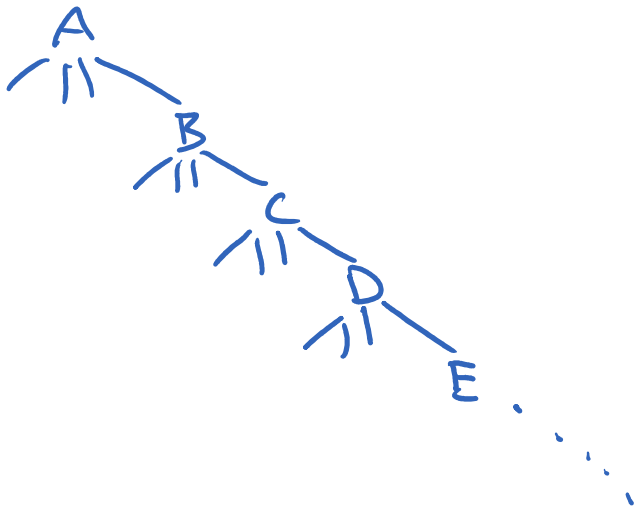




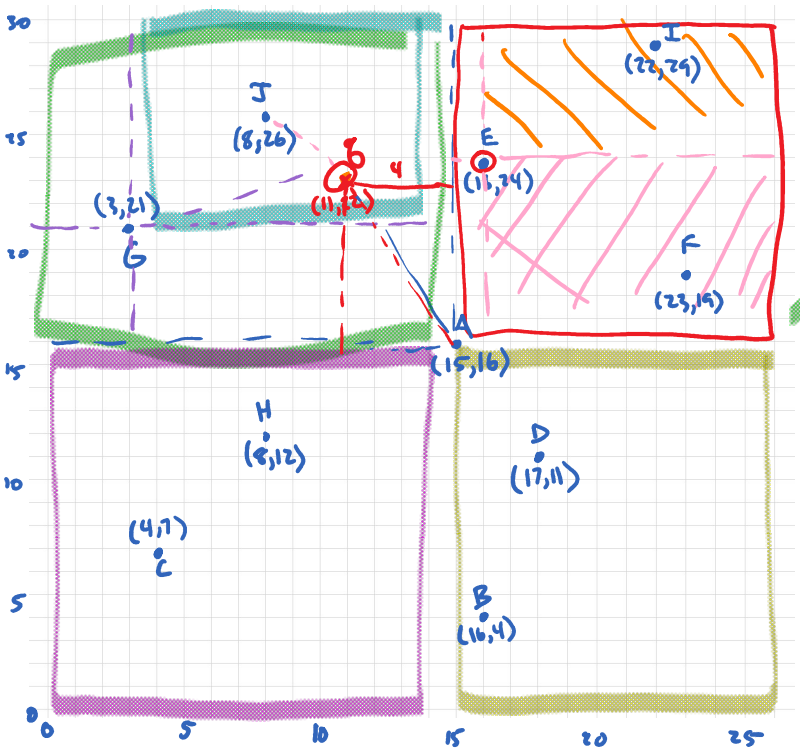
A	15, 16
B	16, 4
C	4, 7
D	17, 11
E	16, 24
F	23, 19
G	3, 21
H	8, 12
I	22, 29
J	8, 26

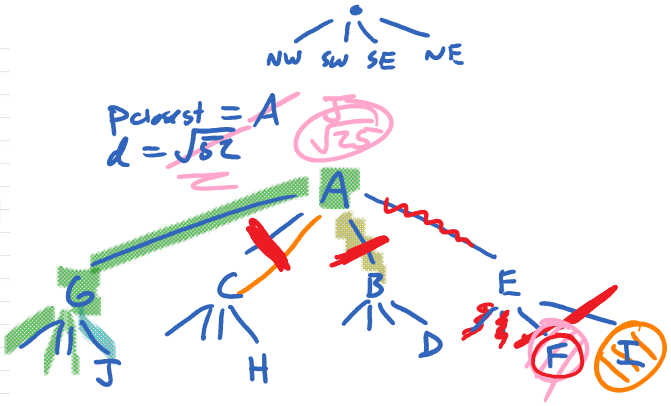
Unbalanced





Nearest neighbor





pclosest, d ← root→p, dist(q, root→p)

pclosest, d ← nn_node(root→nw, pclosest, d) ✓

" "

root→sw, . . . ✓

root→se, . . .

root→ne

if subtree empty ✓
return closest, d

✓ if distance from q to region for current node is > d
return closest, d