

① Congrat class Pteam tryouts!

① Red-Black Trees!

Regular BST w/ extra "rules"

- 1) All nodes are colored red or black
- 2) Root is black
- 3) All red nodes must have black children only
- 4) All external node (null leaf) are black.
- 5) The "black height" of any path from root to external leaf node is the same!

2-4 tree

(10)

same

Black
(10)

(10, 20)

same

(10)^B
 (20)^R

or
(20)^B
 (10)^R

(10, 20, 30)

same

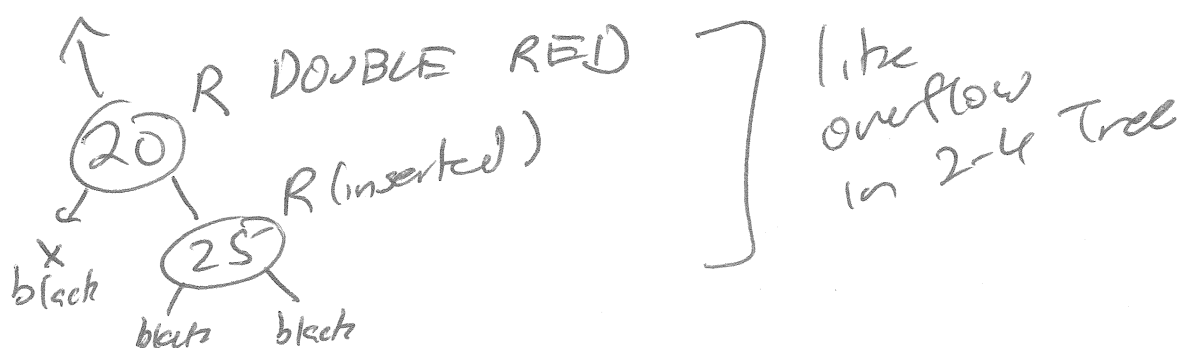
B (20)
R (10) (30)^R

Insert

Regular BST insert, initially color red unless it's the 1st node.

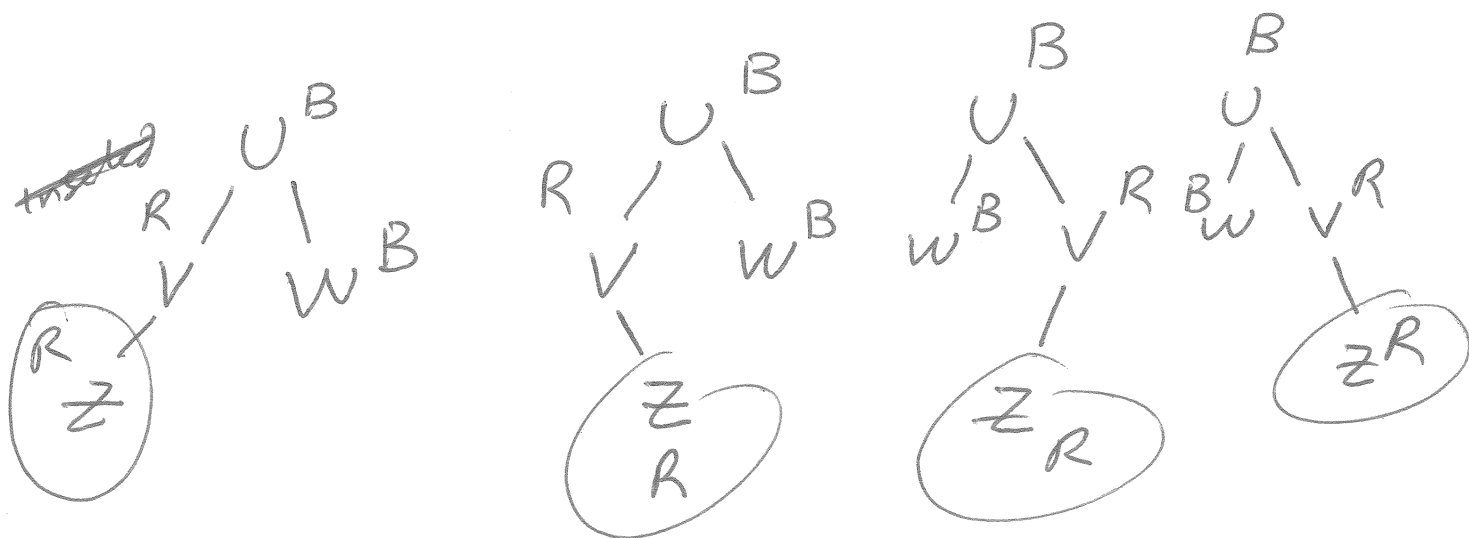
Maintaining black height

But it might screw up: red nodes must have black children rule!



Siblings of inserted node cases:

- ① UNCLE Siblings is black
- 2) UNCLE Siblings is red



Insert - if Case 2 propagates to the root to make it red, just color it black (adds 1 to black height equivalent to case where 2-4 tree grows in height)

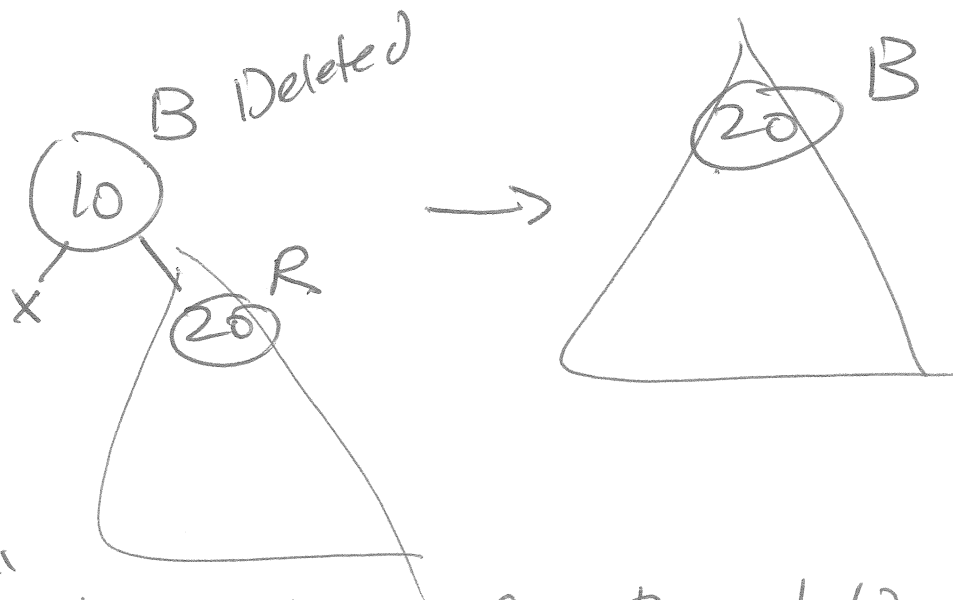
Delete

Only cases we need to consider

- 1) Red leaf node
- 2) Black node with one child (always red)
- 3) Black leaf node

✓ Case 1 - ~~regus~~ regular BST delete suffices.

Case 2 -



" " Promote the color of the child node which replaces deleted node's location in tree always ~~to~~ Red → Black issue DONE

Case 3: Black leaf node $\rightarrow$ Double Black

3-A: Sibling DB is black w/red child

3-B: Sibling DB is black w/2 black children

3-C: Sibling DB is red.

3-A

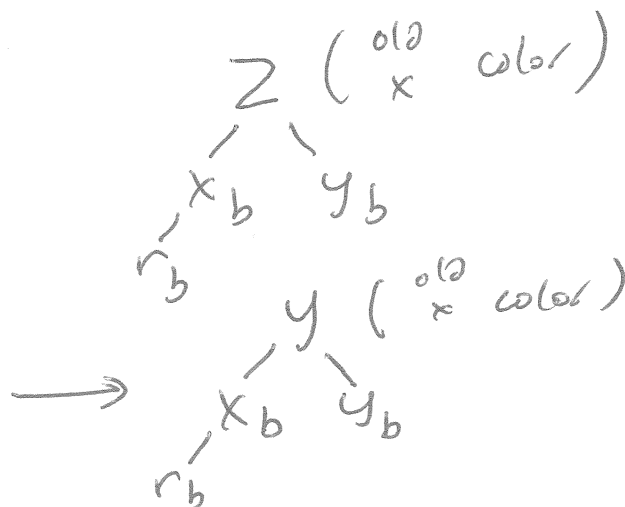
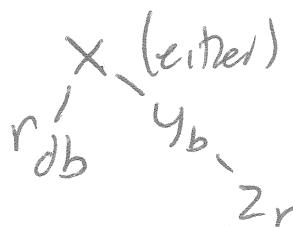
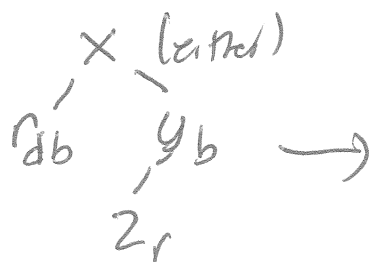
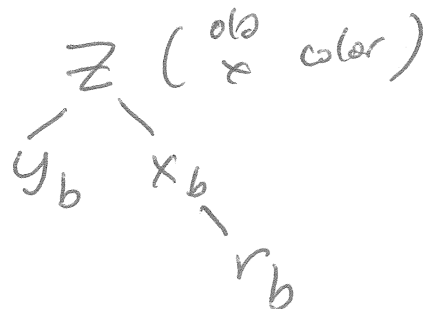
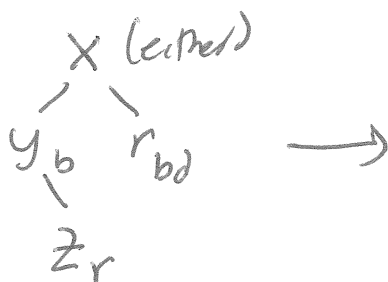
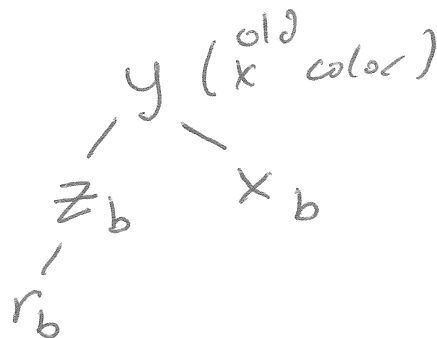
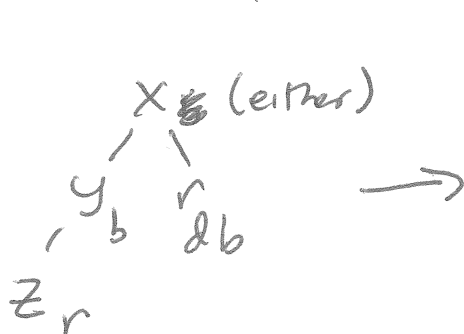
Four pictures

let $y = \text{del node sibling}$

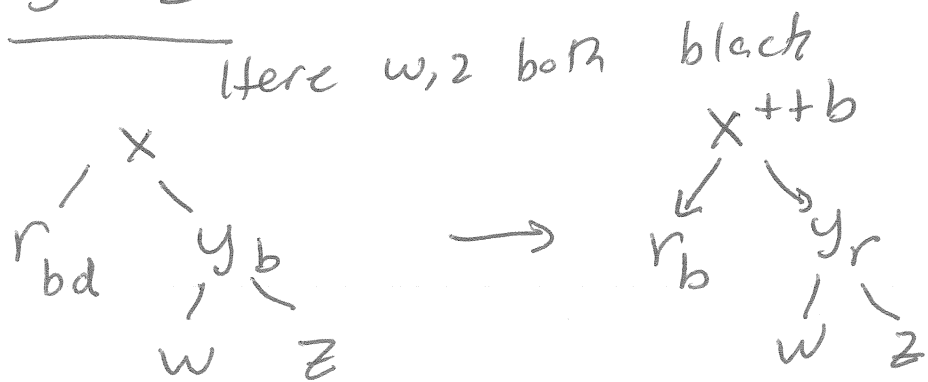
$z = \text{red child } y$

$x = \text{del node parent}$

This fixes issue
as no double
black remains
afterwards.



3-B

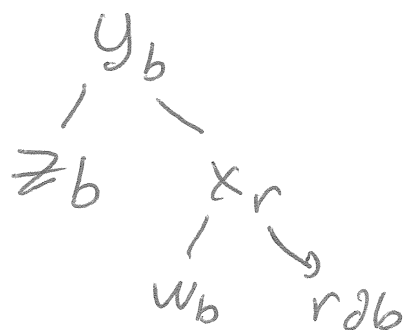
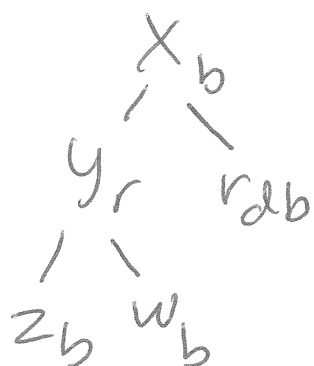


Push black "up"
by subtracting
1 from black
score of $r_b + y$
and adding 1
to black score
of x .

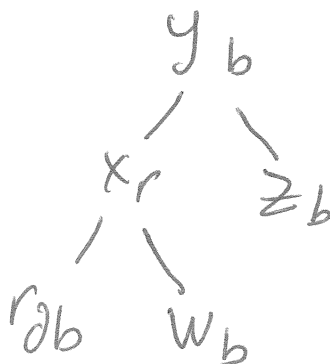
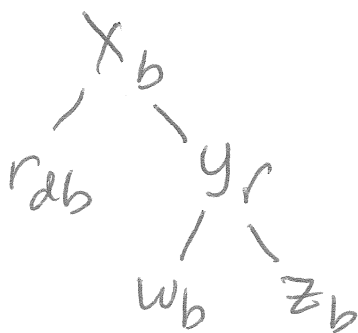
Might make x double black...

If not, done, otherwise fix x + continue!

3-C



Now
Case 3-A or
3-B



Now Case
3-A or
3-B.