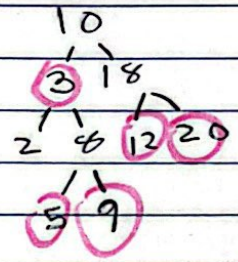
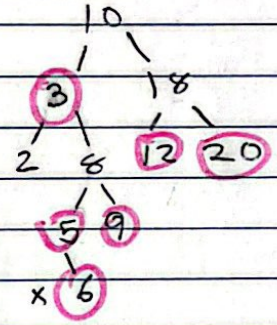
CS280: HW5

Problem 1

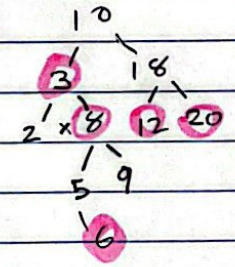
original tree





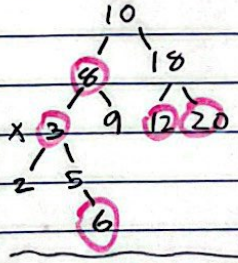
Case 1

- Uncle is red
  - Push blackness from grandparent



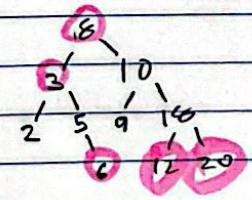
Case 2

- x is a right child, uncle is black
  - Rotate left

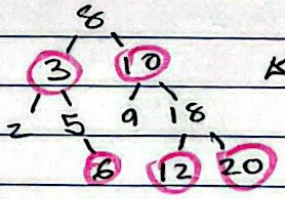


Case 3

- x is left child, uncle is black
  - Right rotate



Recolor so root is black →

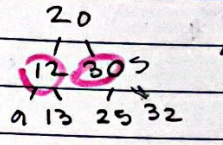


Final tree

CS280 : HW5

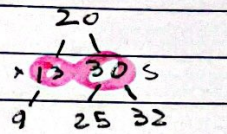
Problem 2

*Delete 12, then 13



← original tree

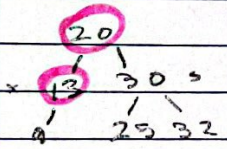
- Replace 12 w/ min in right subtree

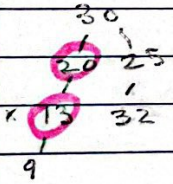


Case 1

S is red

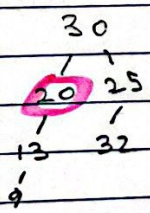
- Switch parent and S colors
- left rotate



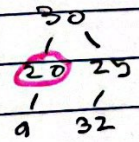


Case 2

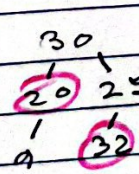
*Recolor x to black



*12 deleted, now 13



- Replace 13 with 9 (only child)



- Recolor to balance

CS280: HW5

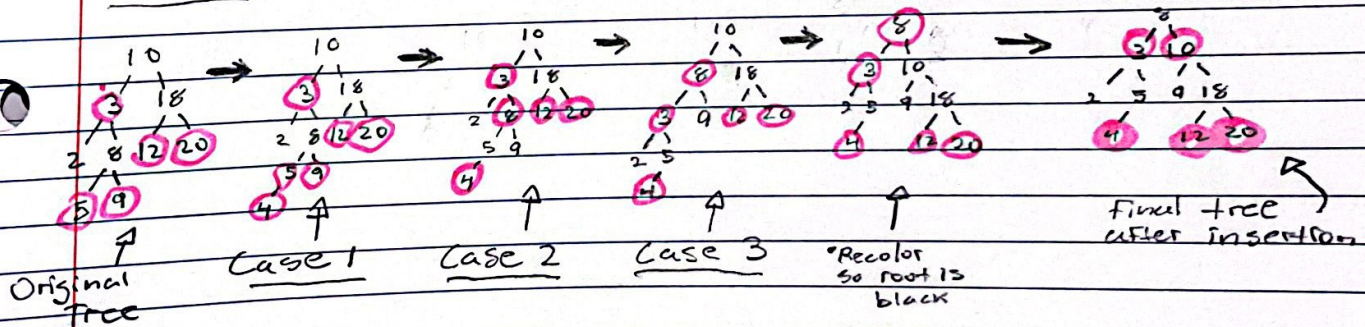
Problem 3

- The result will not necessarily be the original tree after deleting the node, as rebalancing/recoloring a node upon deletion does not always "undo" the rebalancing/Recoloring done upon insertions.

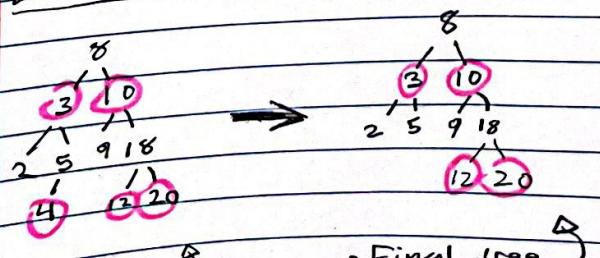
Example (using the tree from problem 1)

- Insert 4, recolor/rebalance, delete 4.

Insertion:



Deletion:



*Since 4 is a leaf node, just delete it.

*Final tree after deletion

CS280: HW8

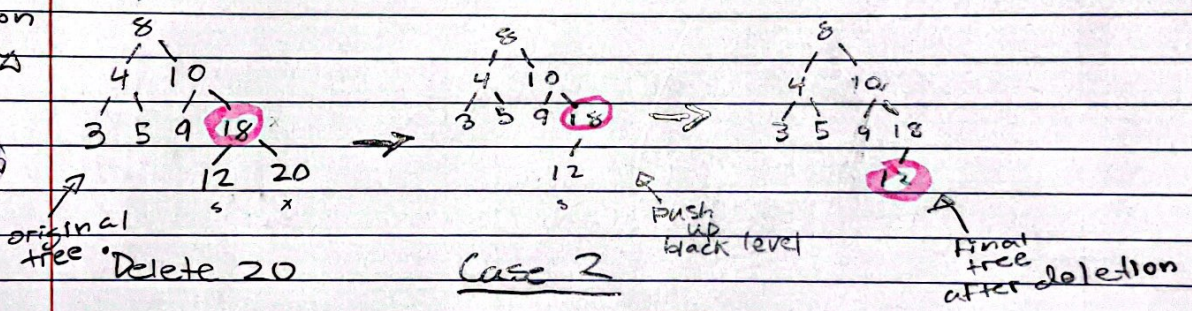
Problem 4

- Deleting a node with no children, rebalancing/Recoloring, then reinserting the same node does not always result in the original tree.

example

- Delete 20 recolor/rebalance

Deletion



Insertion

Insert 20