

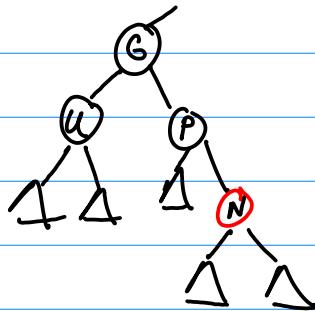
add to a red/black tree

add node as in a BST and color it red  
begin recursive rebalance procedure

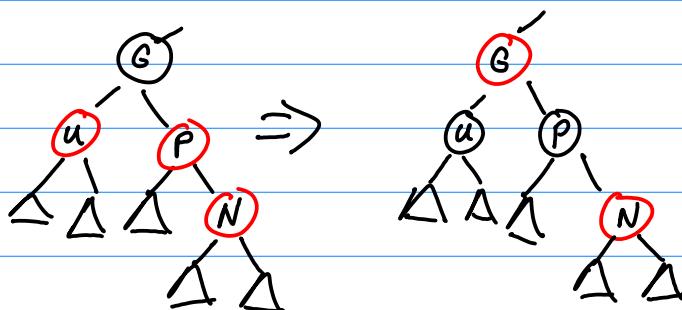
case 1: node is root  
color it black



case 2: node's parent is black  
do nothing



case 3: node's parent and uncle are red  
- color parent and uncle black  
- color grandparent red  
- recursively rebalance grand parent

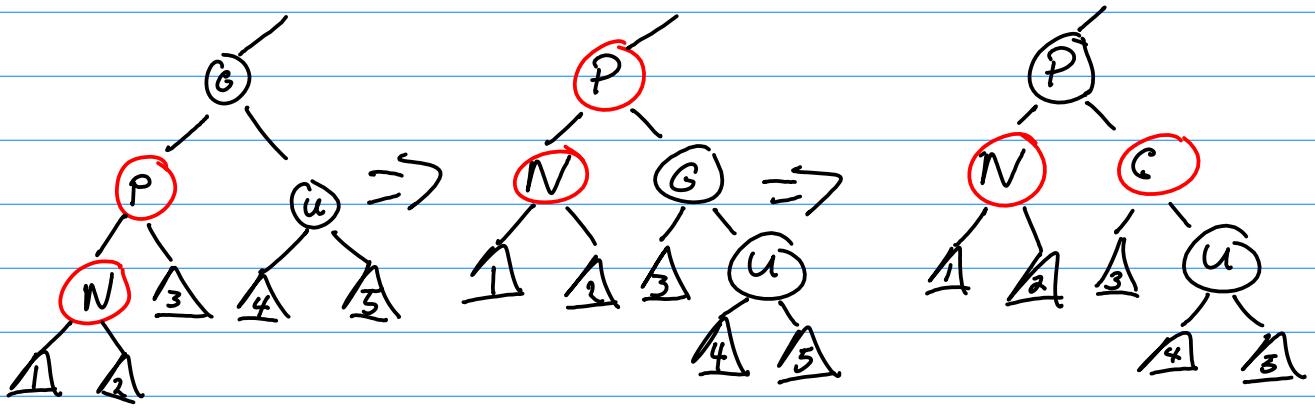


case 4: node's parent is red but uncle is black

sub case 4.1 : parent is left child, node is left

- right rotation around grand parent

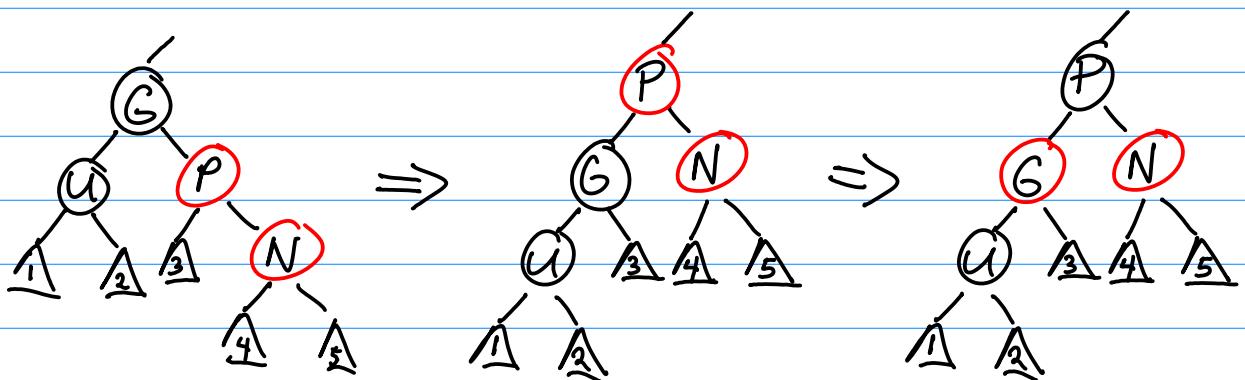
- swap color of parent and new sibling



sub case 4.2 : parent is right child, node is right

- left rotation around grand parent

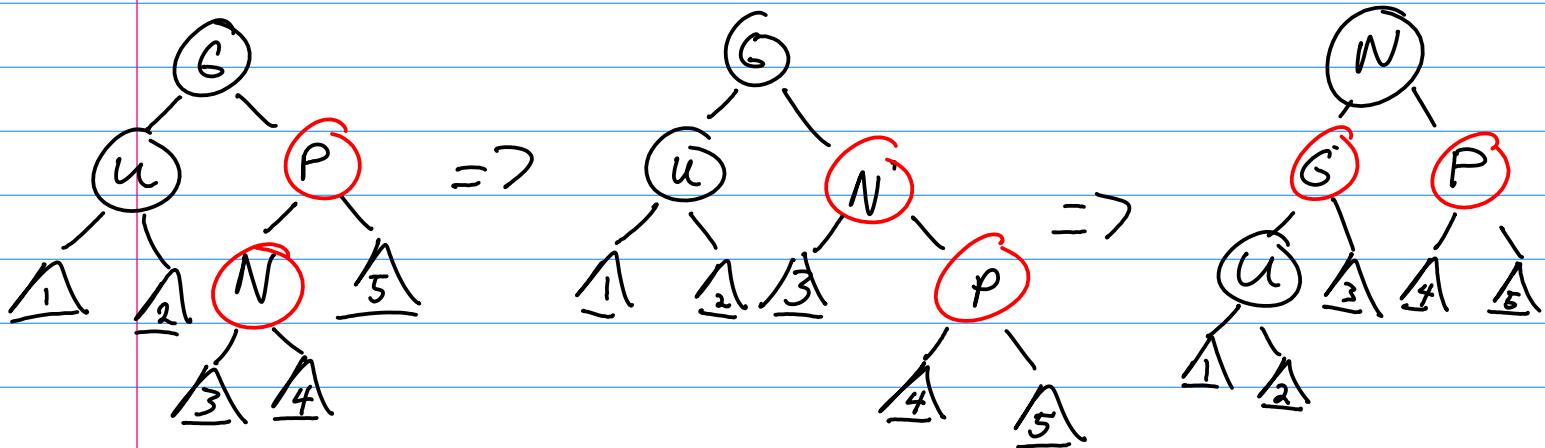
- swap color of parent and new sibling



sub case 4.3 : parent is right child, node is left

- right rotation about parent

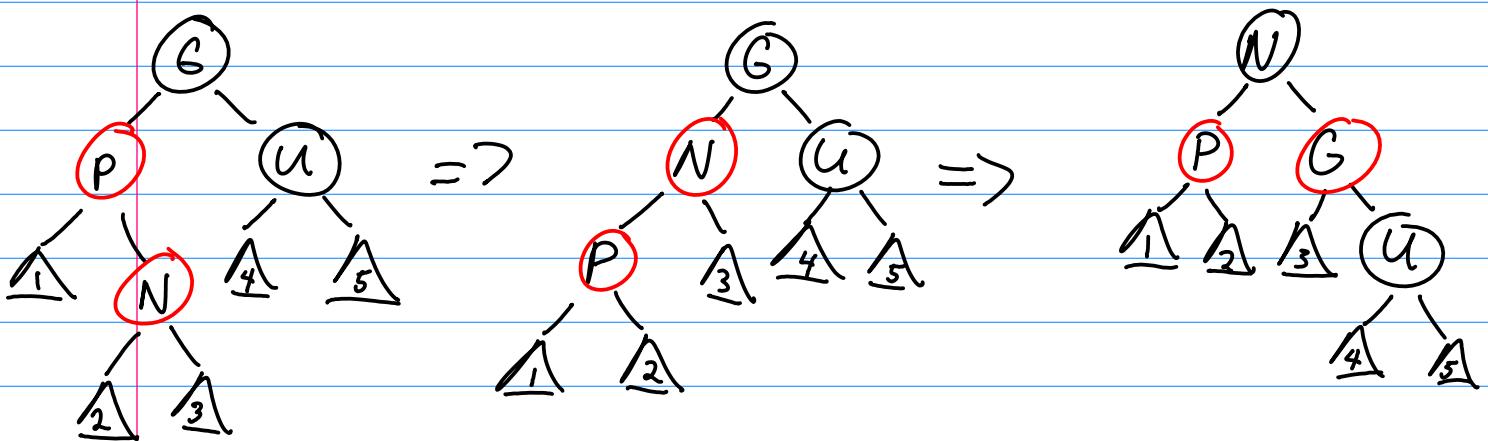
- recursively rebalance parent



sub case 4.4 : parent is left child, node is right

- left rotation about parent

- recursively rebalance parent



Proof that when adding a node to a red/black tree and the new node's parent is black resulting tree is a valid red/black tree

Must show that the resulting tree maintains the properties of a BST and a red/black tree

It does not violate the rules of a BST because the location of the node is based on the rules of a BST

It does not violate the rules of a red/black tree because:

1. The node is colored red, so all nodes are red or black
2. The node is not the root because the proof assumes the node has a parent, so the root is still black
3. The parent is black and the node is red, so no new links between two red nodes are added to the tree.
4. Only one new path is added to the tree after adding the node.

case 1: if parent has no other child then the new path has the same number of black nodes as the path to the parent. Therefore it has the same number of black nodes

as any other path

case 2: if the parent has another child then it must be red and have no children (if it were red then there would be a red child of a red parent, which violates the rules of a red/black tree and if it had children they would have to be black which would mean it must have a sibling with black nodes which it can not if the new node was added as its sibling)  
then the new path has the same number of black nodes because the two paths are identical except for a different red node at the end. Therefore, it has the same number of black nodes.