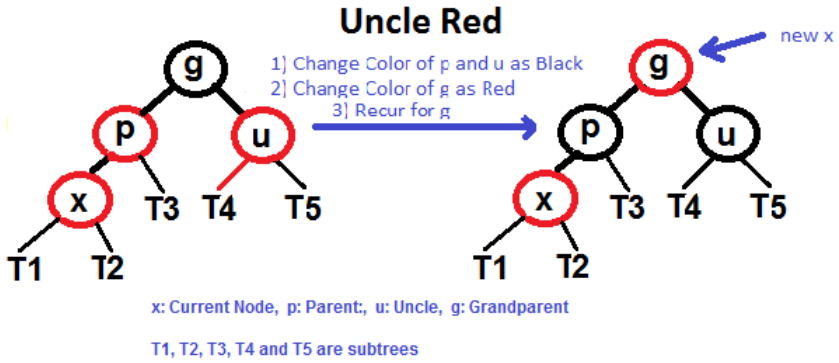
You will are asked to provide a solution to the Red Black Tree Insert problem.

Follow the following principles:

Let x be the newly inserted node.
1) Perform standard BST insertion and make the color of newly inserted nodes as RED.
2) If x is root, change color of x as BLACK (Black height of complete tree increases by 1).
3) Do following if color of x's parent is not BLACK or x is not root.

a) If x's uncle is RED (Grand parent must have been black from property 4)

(i) Change color of parent and uncle as BLACK.
(ii) color of grand parent as RED.
(iii) Change x = x's grandparent, repeat steps 2 and 3 for new x.



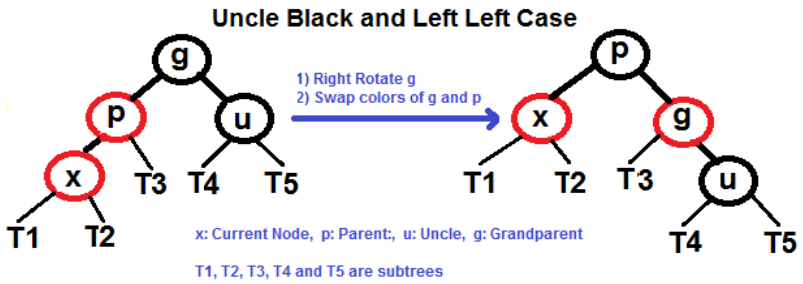
b) If x's uncle is BLACK, then there can be four configurations for x, x's parent (**p**) and x's grandparent (**g**) (This is similar to AVL Tree)

i) Left Left Case (p is left child of g and x is left child of p)
ii) Left Right Case (p is left child of g and x is right child of p)
iii) Right Right Case (Mirror of case a)
iv) Right Left Case (Mirror of case c)

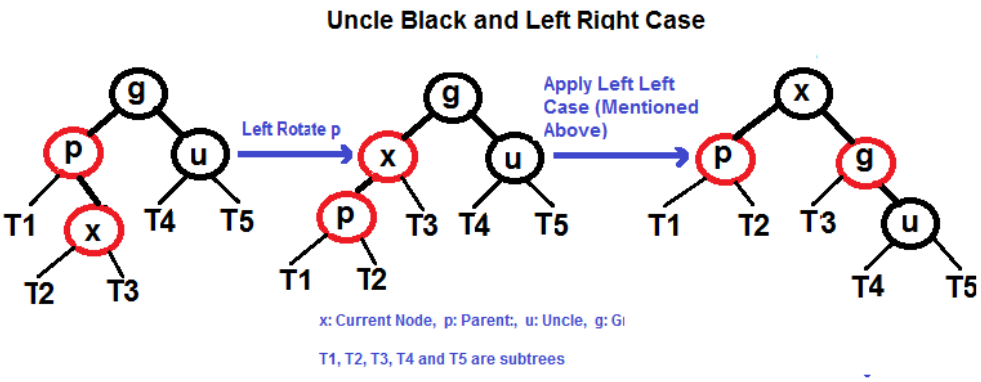
Following are operations to be performed in four subcases when uncle is BLACK.

All four cases when Uncle is BLACK

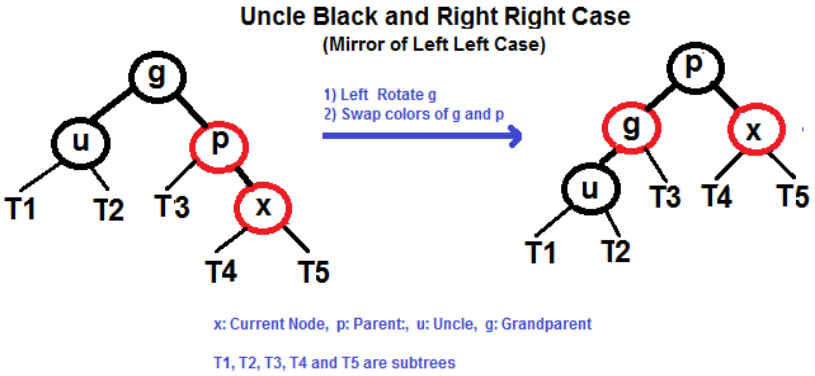
Left Left Case (See g, p and x)



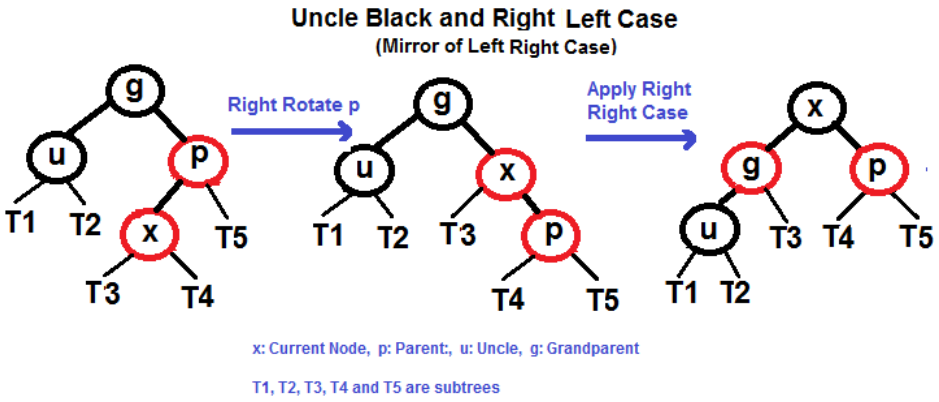
Left Right Case (See g, p and x)



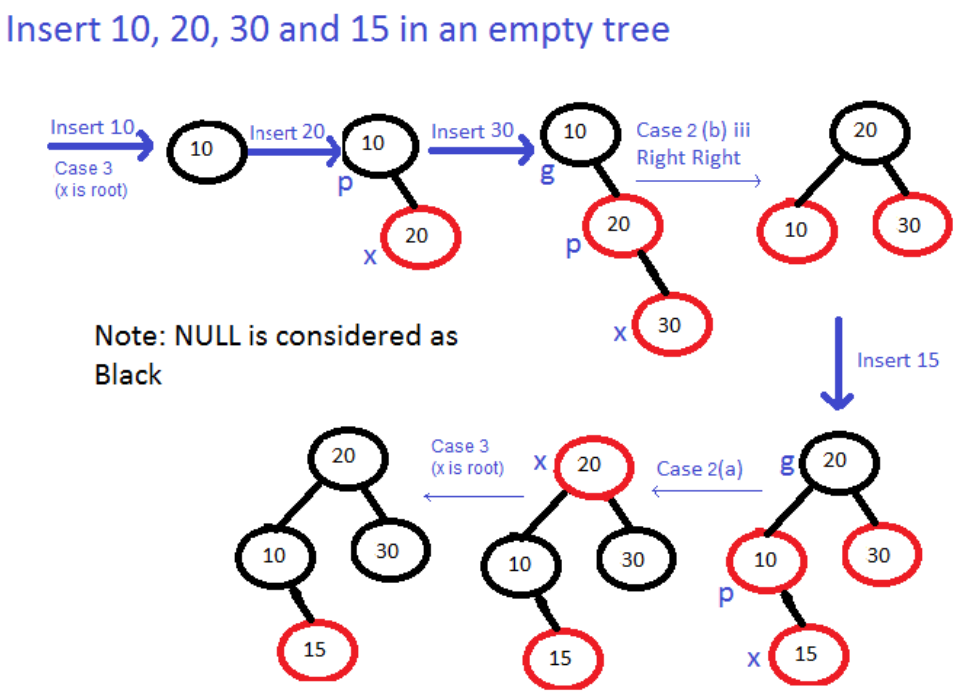
Right Right Case (See g, p and x)



Right Left Case (See g, p and x)



Examples of Insertion



Task 1: Provide code that conducts Red Black Tree Insertion as outlined above.

For this example we draw motivation from: https://www.geeksforgeeks.org/c-program-red-black-tree-insertion/