Red-Black Tree | Set 2 (Insert)

In the previous post, we discussed introduction to Red-Black Trees. In this post, insertion is discussed.

In AVL tree insertion, we used rotation as a tool to do balancing after insertion caused imbalance. In Red-Black tree, we use two tools to do balancing.

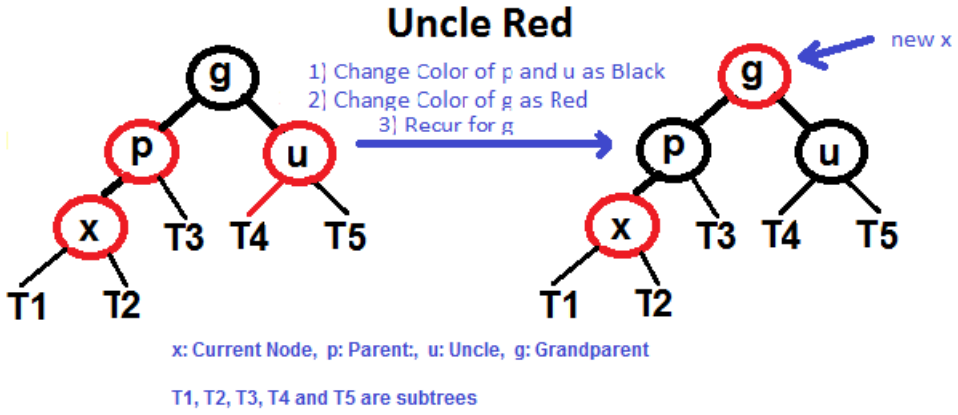
1) Recoloring
2) Rotation

We try recoloring first, if recoloring doesn't work, then we go for rotation. Following is detailed algorithm. The algorithms has mainly two cases depending upon the color of uncle. If uncle is red, we do recoloring. If uncle is black, we do rotations and/or recoloring.

Color of a NULL node is considered as BLACK.

Let x be the newly inserted node.
1) Perform standard BST insertion and make the color of newly inserted nodes as RED.

2) 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.



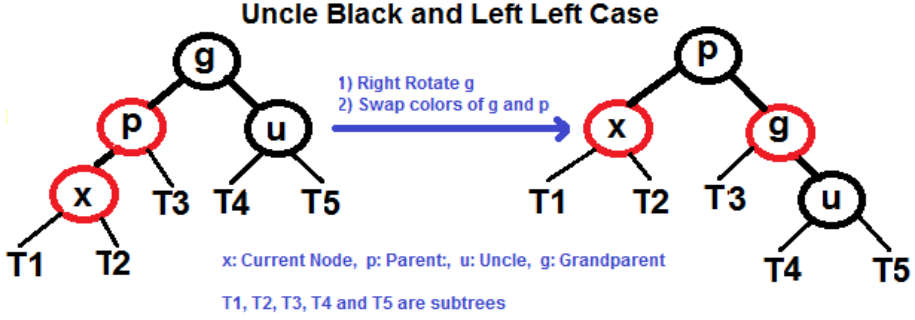
....**b) If x's uncle is BLACK**, then there can be four configurations for x, x's parent (**p**) and x's grandparent (**g**) (This is similar to AVL Tree)
.....**i)** Left Left Case (p is left child of g and x is left child of p)
.....**ii)** Left Right Case (p is left child of g and x is right child of p)
.....**iii)** Right Right Case (Mirror of case a)
.....**iv)** Right Left Case (Mirror of case c)

3) If x is root, change color of x as BLACK (Black height of complete tree increases by 1).

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)

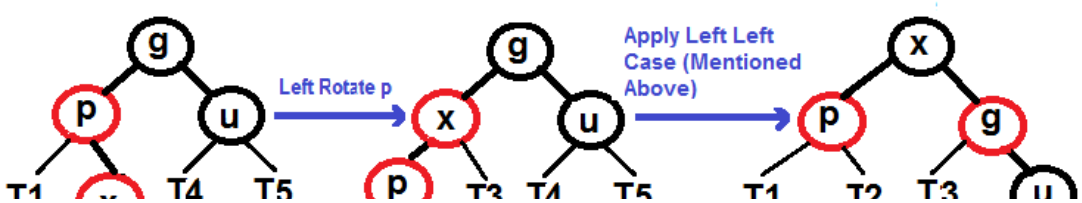
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

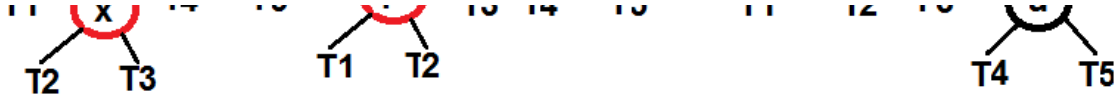
Popular Posts

All permutations of a given string
Memory Layout of C Programs
Understanding "extern" keyword in C
Median of two sorted arrays
Tree traversal without recursion and without stack!
Structure Member Alignment, Padding and Data Packing

Intersection point of two Linked Lists
Lowest Common Ancestor in a BST.
Check if a binary tree is BST or not
Sorted Linked List to Balanced BST

Uncle Black and Left Right Case

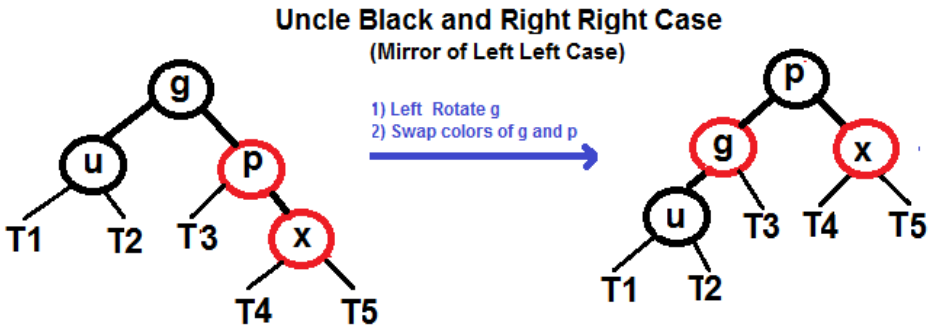




x: Current Node, p: Parent:, u: Uncle, g: Gr

T1, T2, T3, T4 and T5 are subtrees

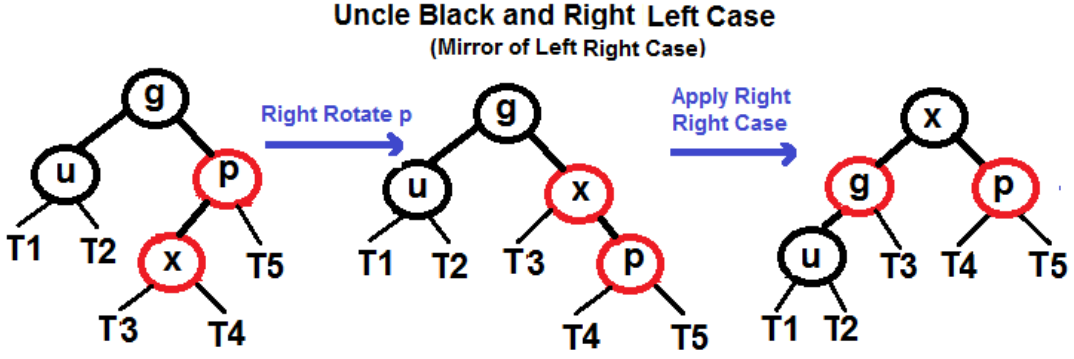
Right Right Case (See g, p and x)



x: Current Node, p: Parent:, u: Uncle, g: Grandparent

T1, T2, T3, T4 and T5 are subtrees

Right Left Case (See g, p and x)

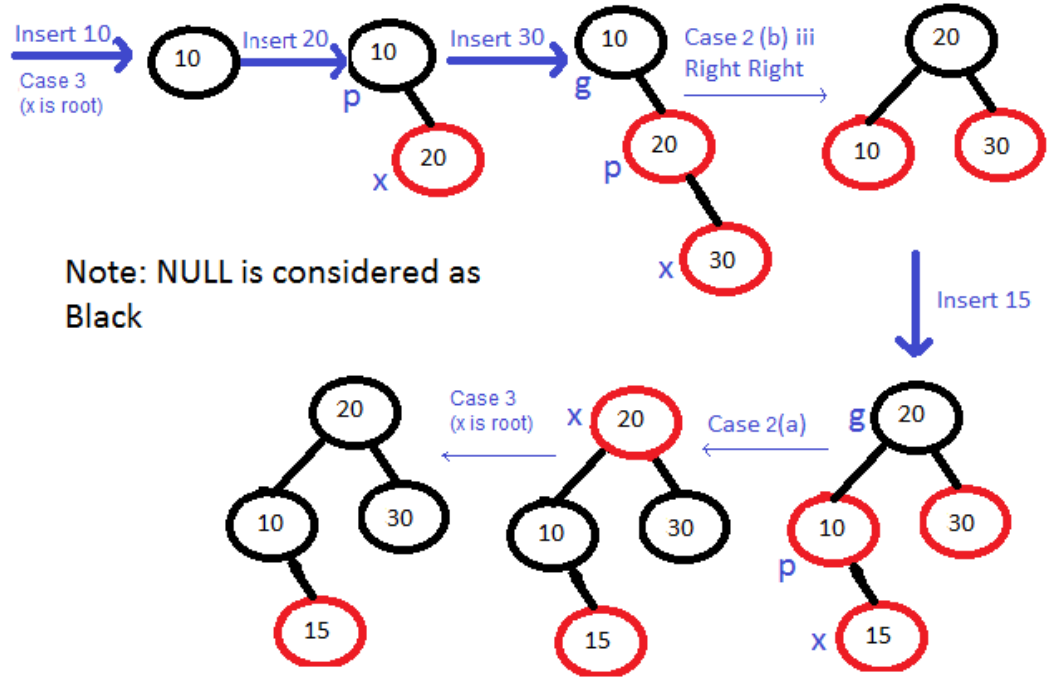


x: Current Node, p: Parent:, u: Uncle, g: Grandparent

T1, T2, T3, T4 and T5 are subtrees

Examples of Insertion

Insert 10, 20, 30 and 15 in an empty tree



We will soon be discussing implementation of above insertion algorithm.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

695

Subscribe

Recent Comments

affiszerv Your example has two 4s on row 3, that's why it...

Backtracking | Set 7 (Sudoku) · 25 minutes ago

RVM Can someone please elaborate this Qs from above...

Flipkart Interview | Set 6 · 45 minutes ago

Vishal Gupta I talked about as an Interviewer in general,...

Software Engineering Lab, Samsung Interview | Set 2 · 45 minutes ago

@meya Working solution for question 2 of 4f2f round....

Amazon Interview | Set 53 (For SDE-1) · 1 hour ago

sandeep void rearrange(struct node *head) {...

Related Tpoics:

- Print a Binary Tree in Vertical Order | Set 2 (Hashmap based Method)
- Print Right View of a Binary Tree
- Red-Black Tree | Set 3 (Delete)
- Construct a tree from Inorder and Level order traversals
- Print all nodes at distance k from a given node
- Print a Binary Tree in Vertical Order | Set 1
- Interval Tree
- Check if a given Binary Tree is height balanced like a Red-Black Tree

Writing code in comment? Please use **ideone.com** and share the link here.

6 Comments GeeksforGeeks

Sort by Newest

Given a linked list, reverse alternate nodes and append at the end · 2 hours ago

Neha I think that is what it should return as, in...

Find depth of the deepest odd level leaf node · 2 hours ago

kaushik Lele • 19 days ago

I ran RedBlackTreeViewer.java; mentioned below. But code is not correct. Wh
4,7,17; it showed a tree with 4 as root and 12 as left node !! that is wrong. So

Reply • Share ›

kaushik Lele • 19 days ago

You can take complete Java code. Take java classes from here. As they are ir
http://cs.lmu.edu/~ray/notes/r...
http://cs.lmu.edu/~ray/notes/b...
http://cs.lmu.edu/~ray/notes/b...

Run RedBlackTreeViewer.java

Reply • Share ›

kaushik Lele • 19 days ago

Very well Explained. Keep these rules handy and view this animation.
http://www.youtube.com/watch?v...

After every insertion; apply these rule and guess what should be behaviour. Yo

Reply • Share ›

robinhood • a month ago

very well xplained.:)

Reply • Share ›

sachi059 • 2 months ago

Good Explanation :-)

Reply • Share ›

Anuj • 2 months ago

First time ever, I understood insertion in red-black tree. Thanks.

2 ^ | v • Reply • Share ›

Subscribe Add Disqus to your site