Red Black Tree :- (also follow BST rules)

This data structure requires an extra one-bit color field in each node.

Properties :-

(1) It is a self Balancing BST

(1) Every node has a color either Red or Black

(2) Root is always Black & whenever you insert New Node, it is Red always.

(3) Every leaf which is NIL is black

(4) If Node is red then its both children must be black or
If a node is red, then its parent must be black.

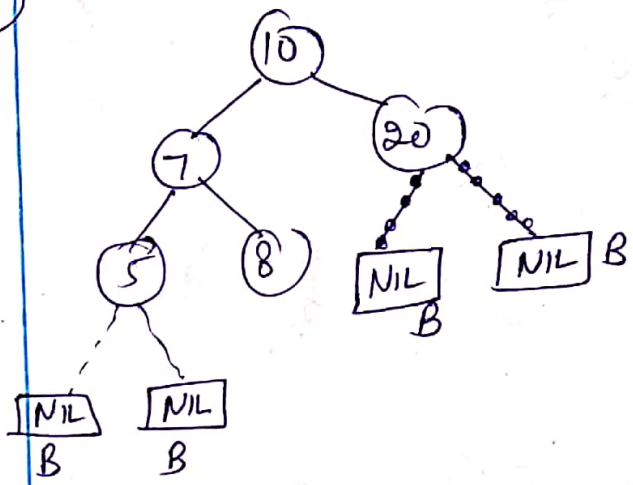
(5) For x node, all paths from the x node to descendant leaves contain the same no. of black nodes.

14/8/18

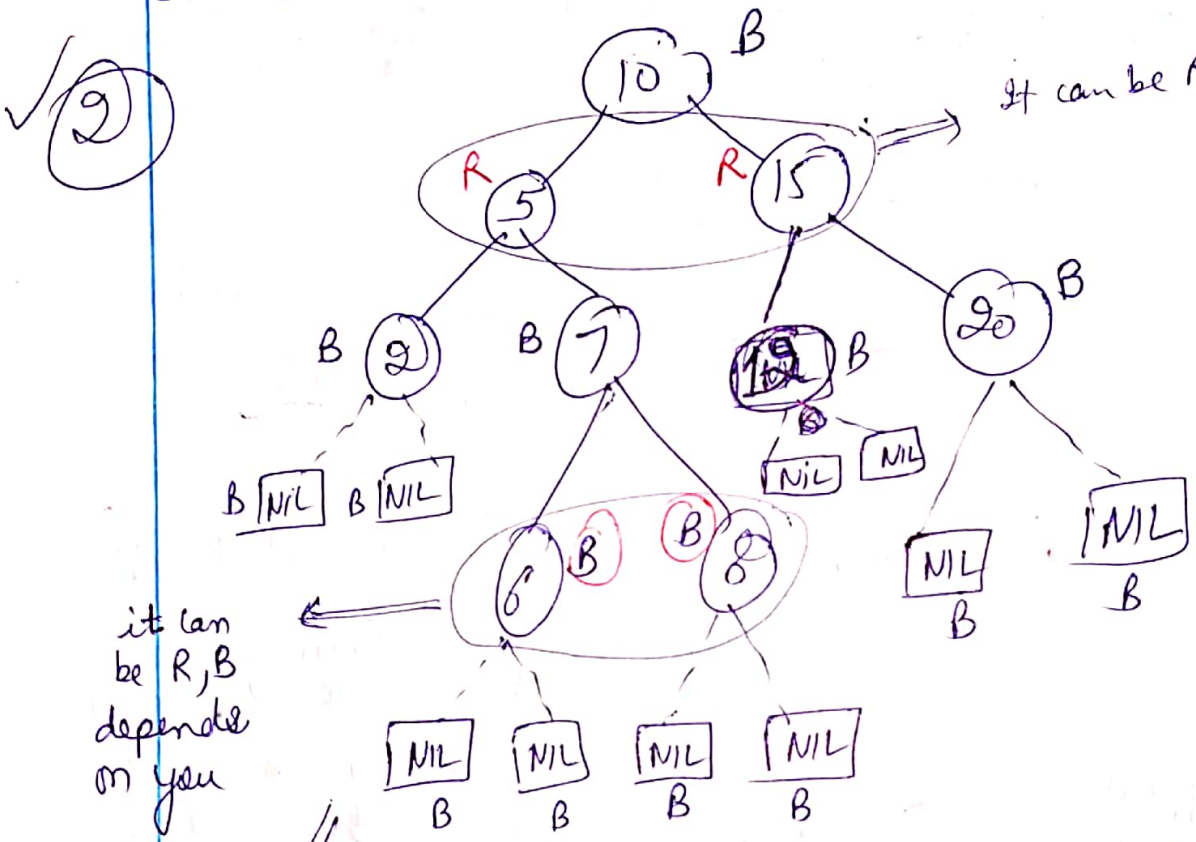
Example:-

⇒ Nil Nodes consider as External Nodes & rest all nodes are internal Nodes.

(1)



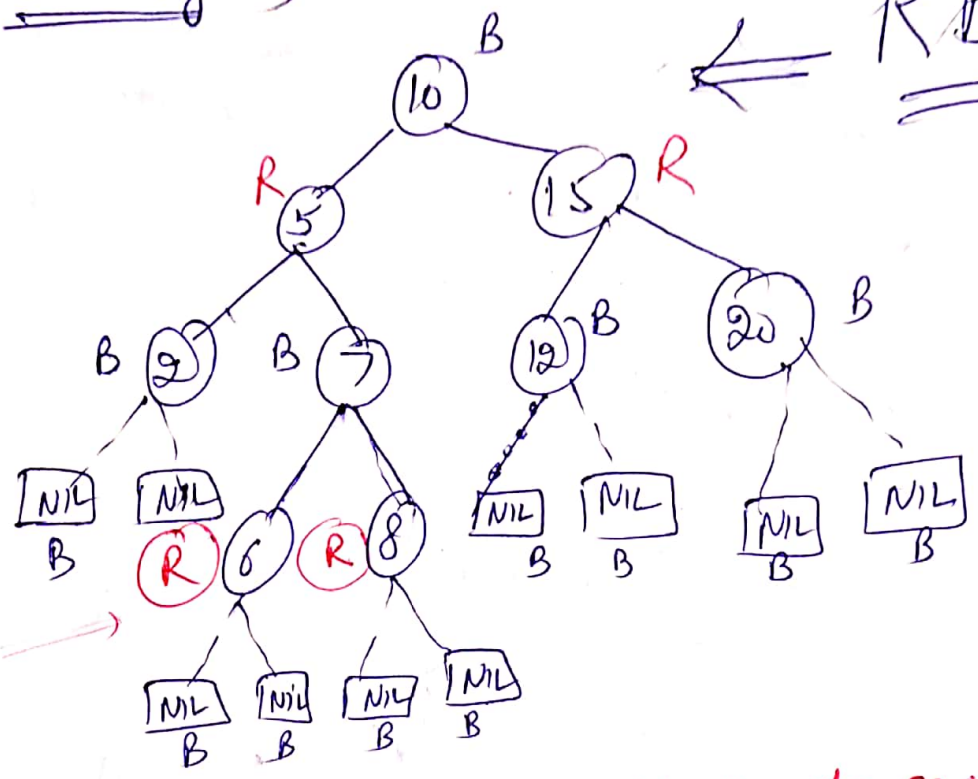
(2)



* Not Count Nil Node while count Black node in each Path ∴ its a External Node.

Because this example each part is not having same no. of Black node ∴ It is not Red Black Tree.

Now Change ⇒



Now, No. of Black at each Path is equal.

NOTE:- Is every AVL Tree is the Red Black Tree?

(Strictly Height Balanced Tree) (Roughly Height Balance Tree)

⇒ Suppose if a Tree is AVL Tree & you color it as Red & Black acc. to the rule than we call AVL Tree as Red Black Tree.

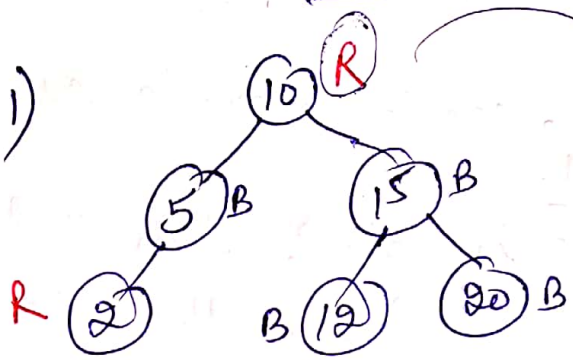
So, we can say AVL Tree are subset of Red Black Tree.

⇒ But if Tree is Red Black Tree than it is not Compulsory/True that it could be a AVL Tree.

Question :-

Example :- Check Whether it is RBT or Not ?

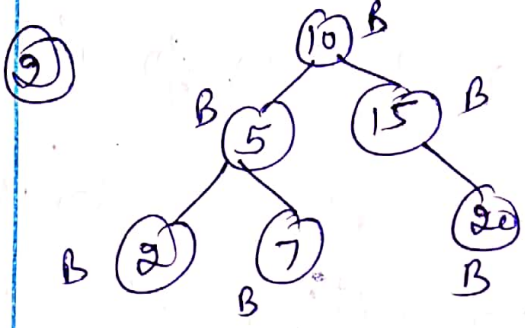
1)



⇒ Not RBT

Root Property of Root Node violates than need not to check other property. (i.e. Root → Black)

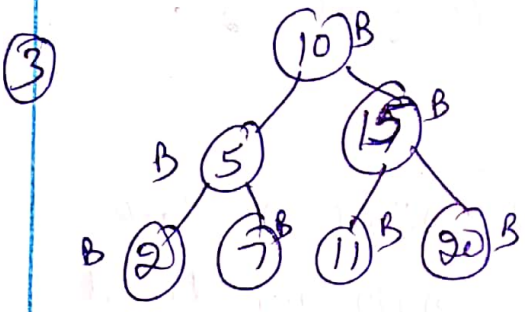
2)



⇒ Not RBT

Each Path from node to any of its descendant ✗

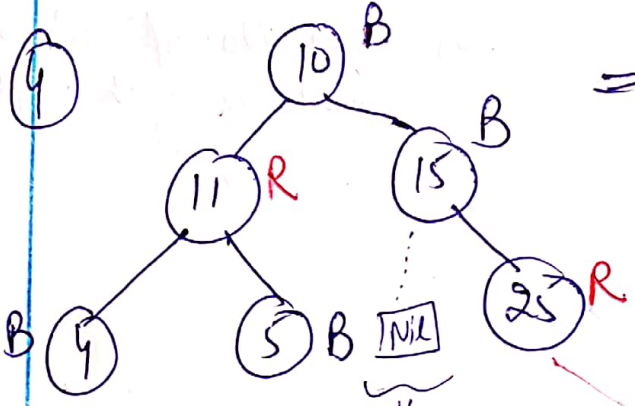
3)



⇒ RBT

(Even we don't have any Red Node Still it is Red Black Tree).

4)



⇒ X RBT (∵ No BST Property)

Rather than draw Nil everywhere just draw {left of 15} cases

Non-graded

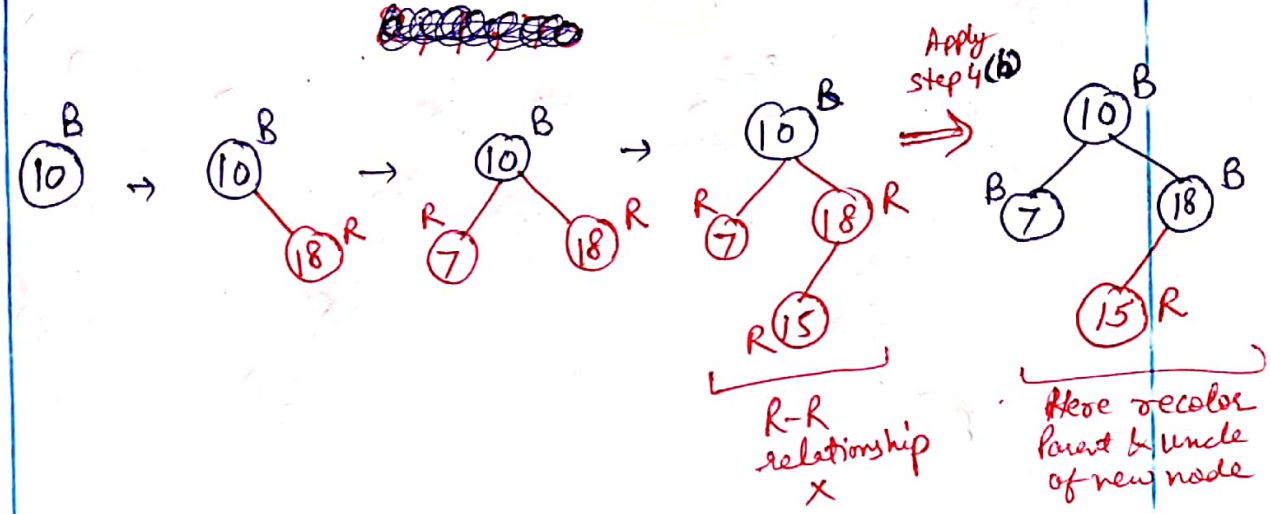
Algorithm of Insertion of Node in RB:

1. If tree is empty, create newnode as root node with color Black
2. If tree is not empty, create newnode as leaf node with color Red.
3. If parent of newnode is Black then exit.
4. If " " " " Red, then check the color of parent's siblings of newnode.
   a) if color is black or null then do suitable rotation & recolor
   b) if color is Red then recolor & also check if parent's parent of newnode is not root node then recolor it & recheck.

NOTE:-
- Root → Black
- No two adacent red nodes
- Count No. of Black nodes in each path is equal.

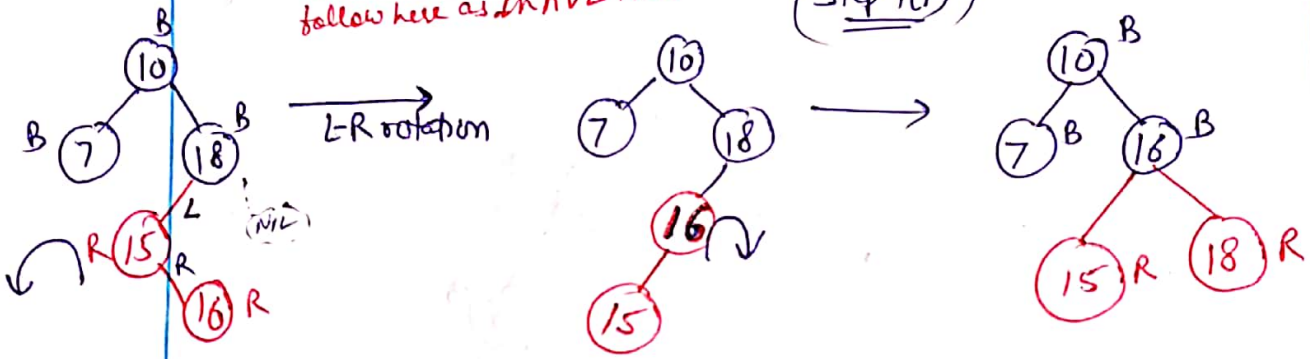
11) RBT ir Insertion in RBT

Example ir 10, 18, 7, 15, 16, 30, 25, [illegible]

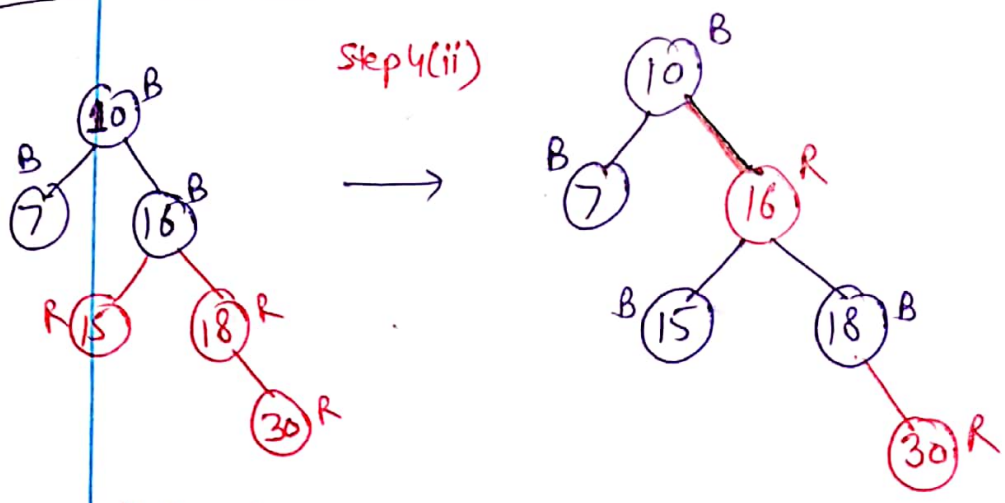


Same Rotation, follow here as in AVL Tree

(Step 4(i))



Step 4(ii)



R-R Conflict.

Step 4(i)