Red-Black Tree [BST with following properties]

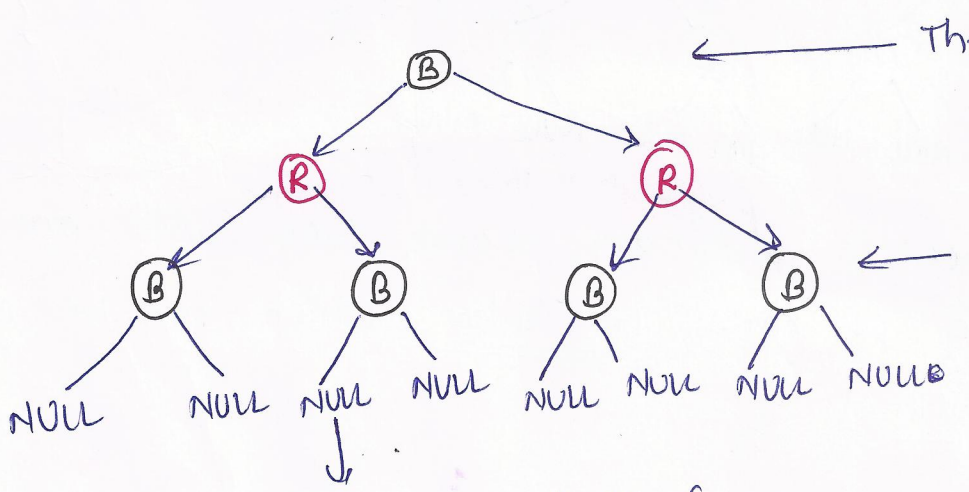
Properties of red black tree

The nodes of a red-black tree are either red or black.

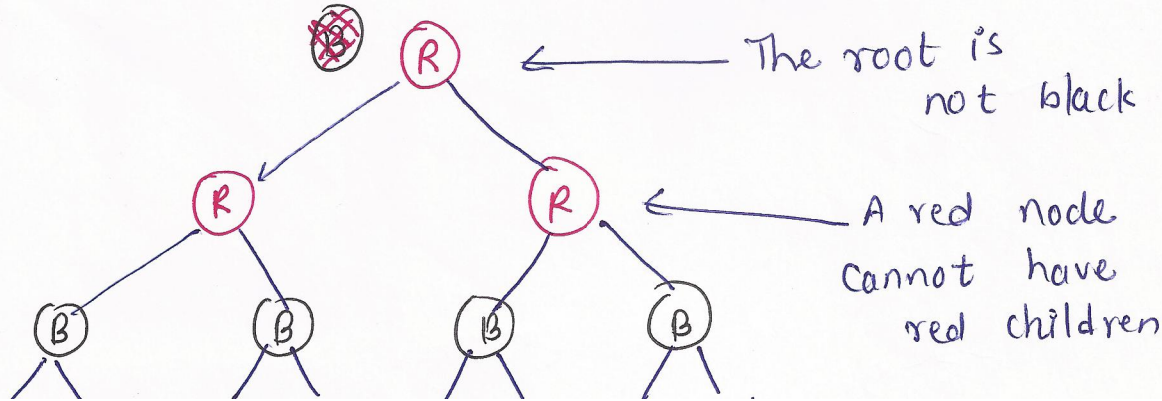
1. The root of the tree is always black.
2. A black node can have a black or a red child.
3. A red node cannot have a red child. It can only have a black child.
4. The black depth of a terminal node is the number of black nodes encountered while travelling from the terminal node to the root.
5. The black depth of a terminal node is always same.

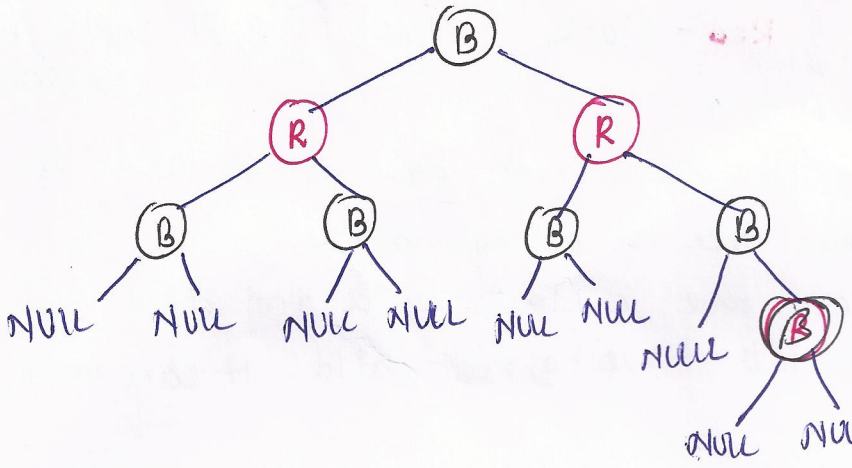
Black depth :- The number of black nodes from the terminal to the root is called the black depth of the node.

6. The leaves of a red-black tree would always be a NULL node.
7. Each black or red node last in the hierarchy will have NULL nodes as children.



a) An example of a red-black tree



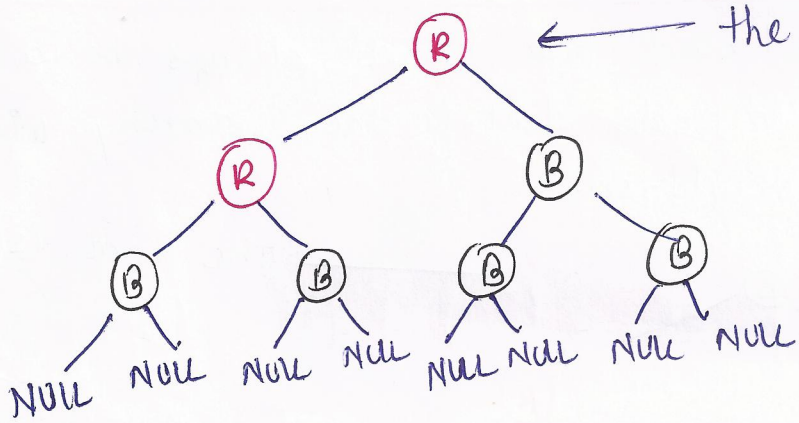


The black depth of this node & its sibling is 3, whereas for the other nodes it is 2.

c) An example of a tree which is not red-black tree

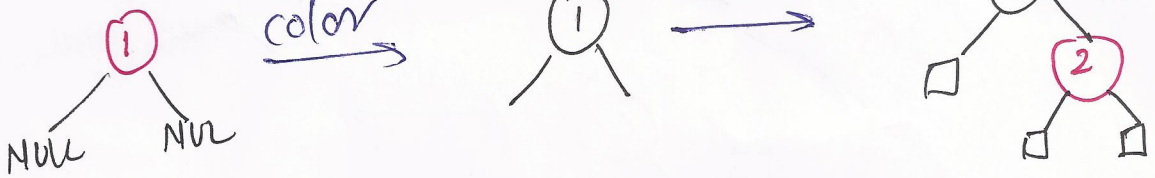
Double red problem

The case wherein the child of a red node is a red node is called the double red problem.

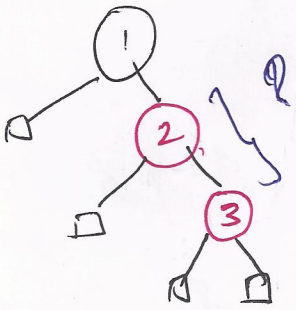


A red-black tree is a binary search tree with one extra bit of storage per node: its color which can be either RED or BLACK

1 2 3 4 5 6



insert 3

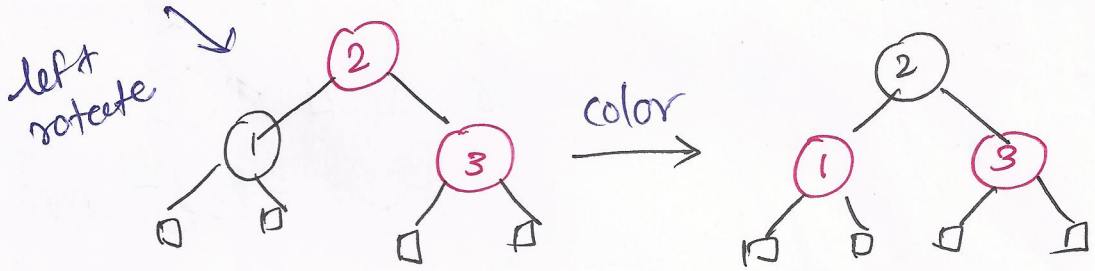


Q How to fix this?

this

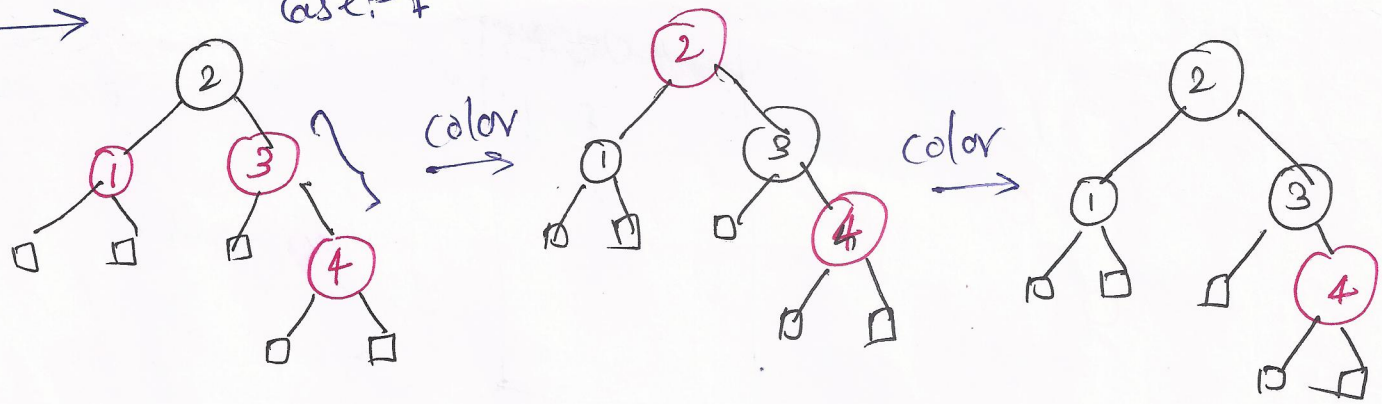
Case:-3 [Uncle is black & insertion is right]

left rotate around the grandparent

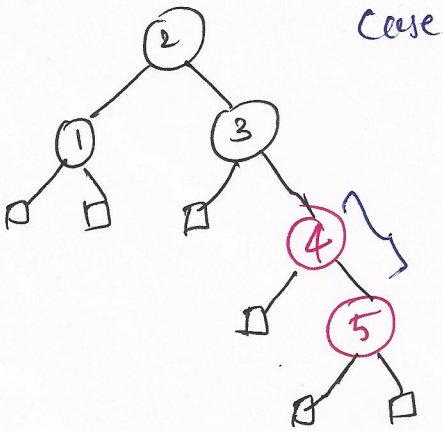


insert 4

Case:-4

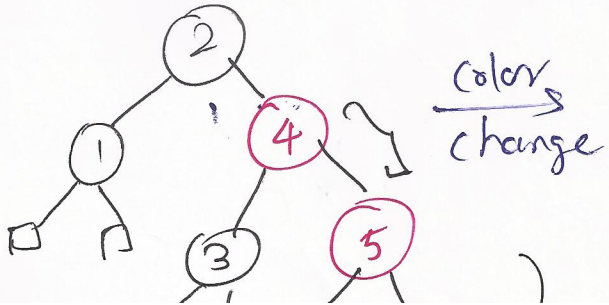


insert 5

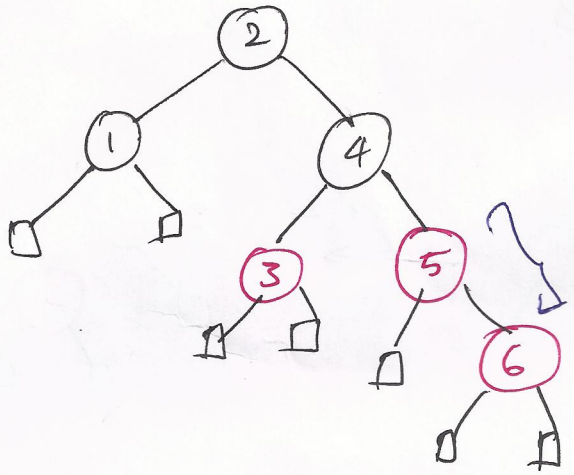


Case 3 [Uncle is black & insertion is at right]

do left rotate

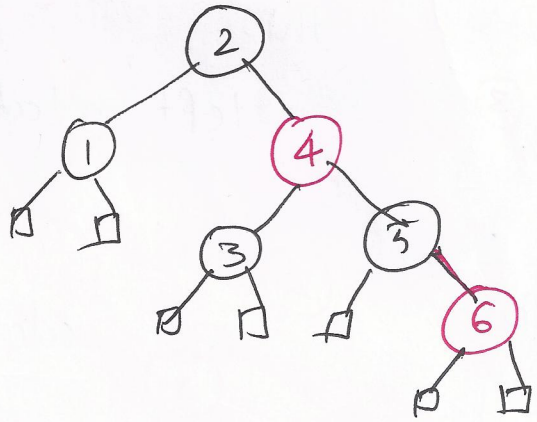


insert 6 →



Uncle is red Case 1

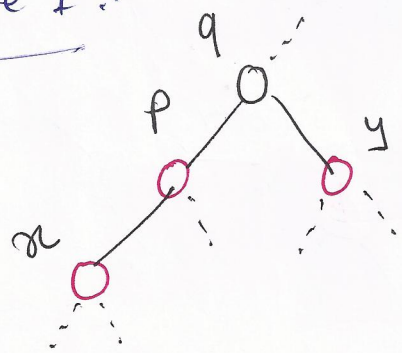
Color →



Black height of this tree is 3

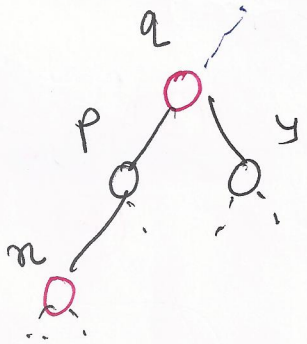
Violations

Case 1 :-

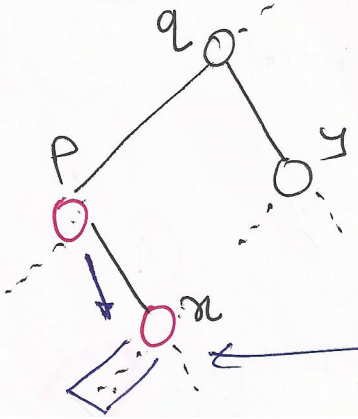


Uncle is Red

Fix : color

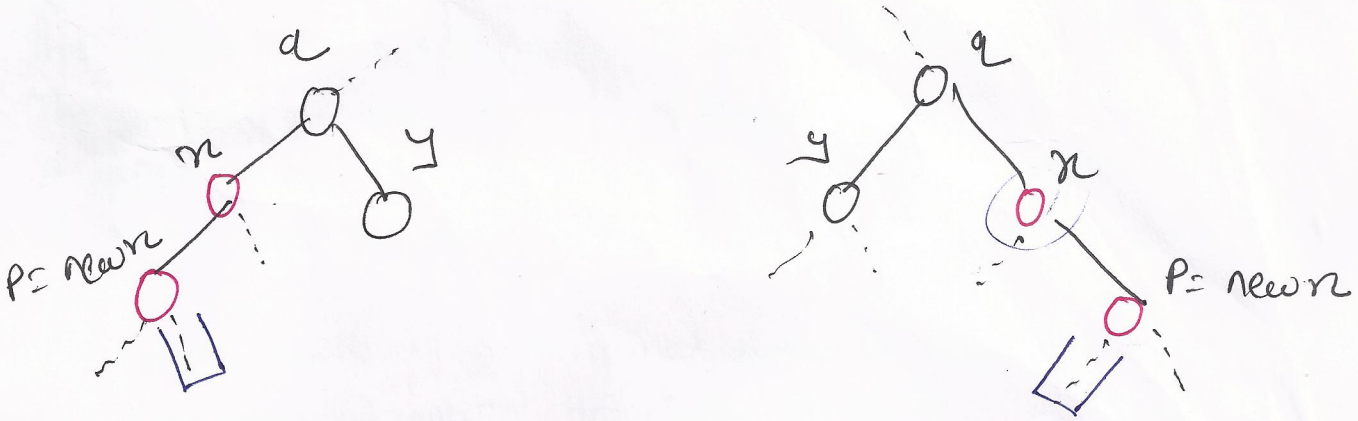


Case 2 :- Uncle is Black Fix : rotate (Parent)



violating node is a right child of a left child

to fix this rotate around the parent node.



Convert to case 3:

uncle is Black

fix rotate (grand parent)

↓

color