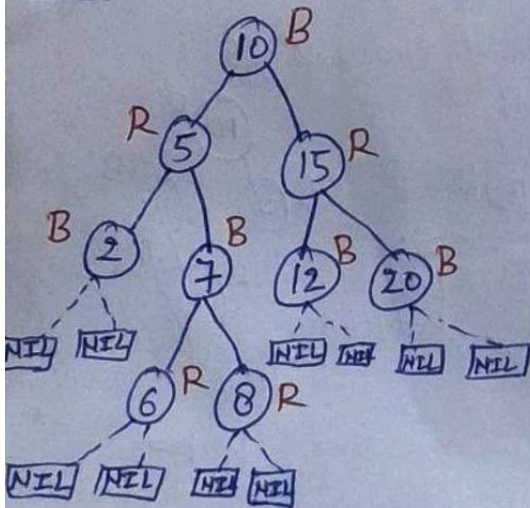# Red-Black Trees

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. By constraining the node colors on any simple path from the root to a leaf, red-black trees ensures that no such path is more than twice as long as any other, so that the tree is approximately balanced.

A red-black tree is a binary tree that satisfies the following red-black properties: → Red-Black tree is a self-balancing BST.
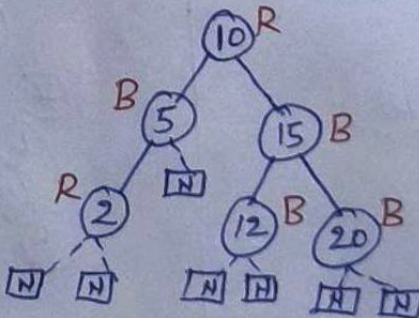
1. Every node is either red or black.

2. The root is black.

3. Every leaf (NIL) is black.

4. If a node is red, then both its children are black.

5. For each node, all simple paths from the node to descendant leaves contain the same number of black nodes.
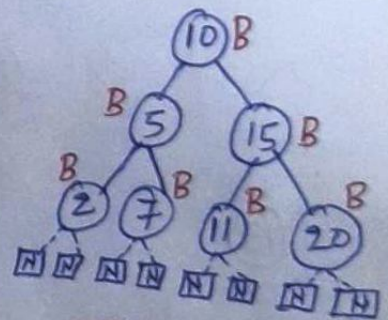
EX-1
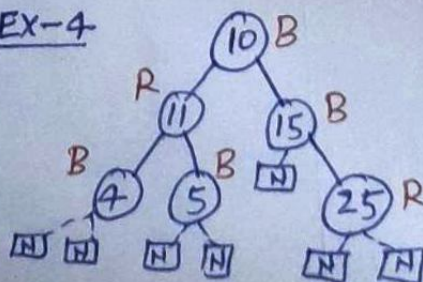

R-B Tree (✓)

EX-2

R-B Tree (X)

(Violate Propery 2)

EX-3

R-B Tree (✓)

Even not a single red color node in a tree.

EX-4

R-B Tree (X) Even though it follows all the propes of R-B tree but it's not a BST.

1

Insertion in Red-Black Tree

Algorithm:

① If tree is empty, ~~then~~ create new node 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 Parent of newnode is 'red' then check the color of parents sibling of new node: _or (uncle)_

  ⓐ If color is black or null then do suitable rotation & recolor.

  ⓑ If color is red then recolor both parent and sibling & also check if parent's parent of newnode is not root node then recolor it & recheck.

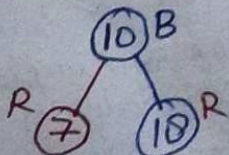Array: 10, 18, 7, 15, 16, 30, 25, 40, 60, 2, 1, 70
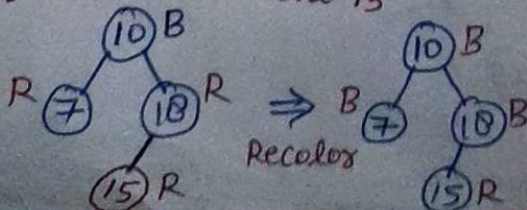
Step 1: Insert element 10
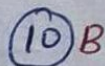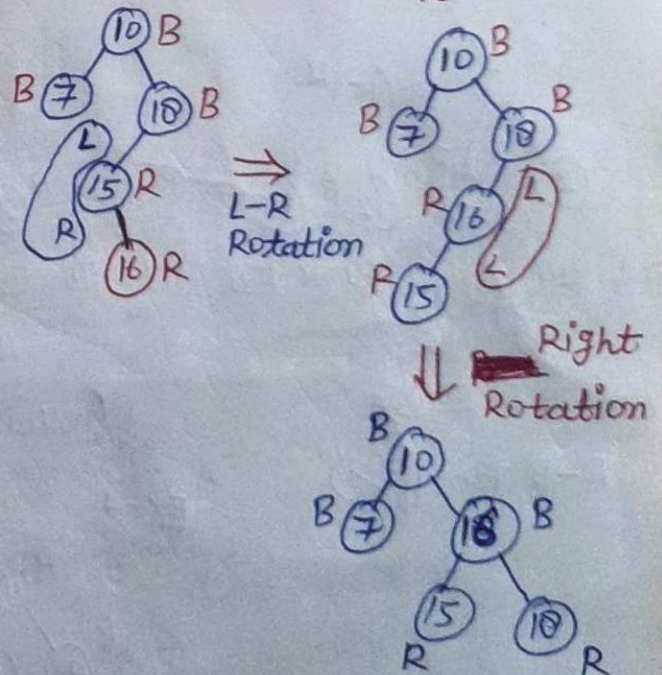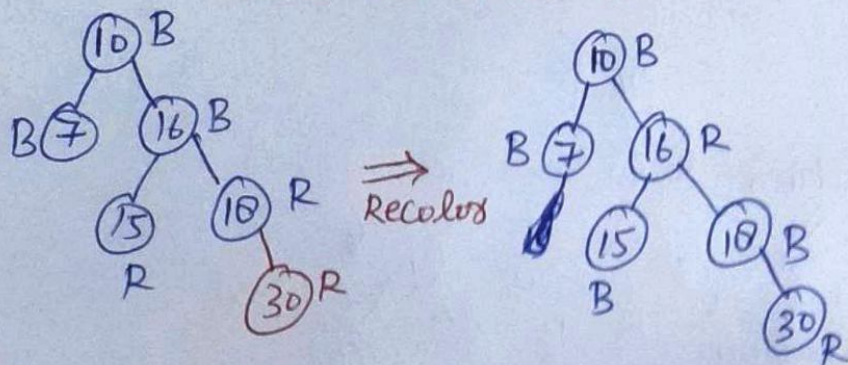

Step 2: Insert element 18
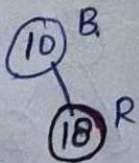
Step 3: Insert element 7
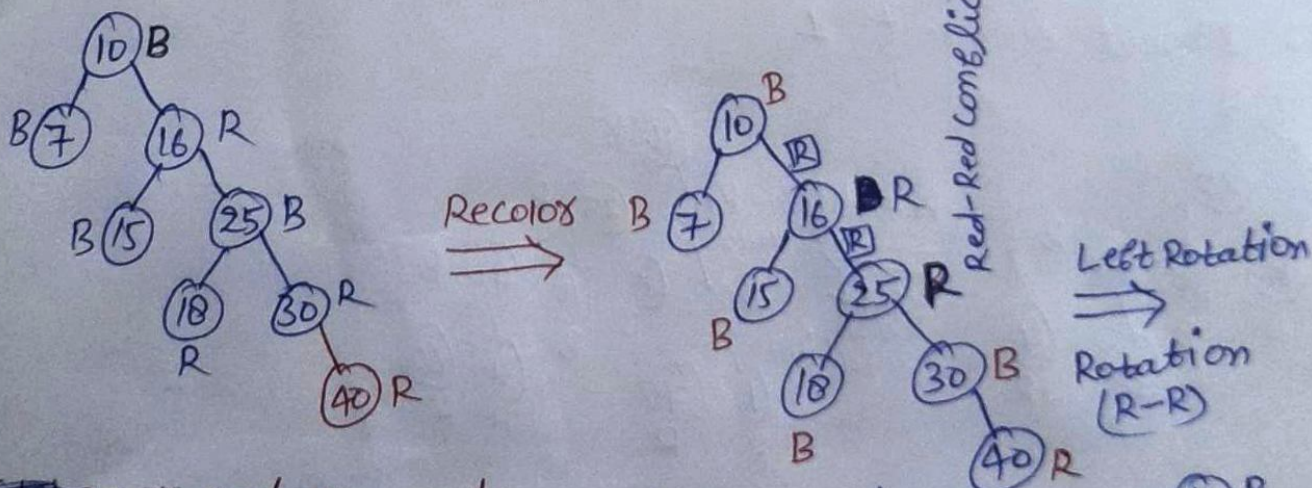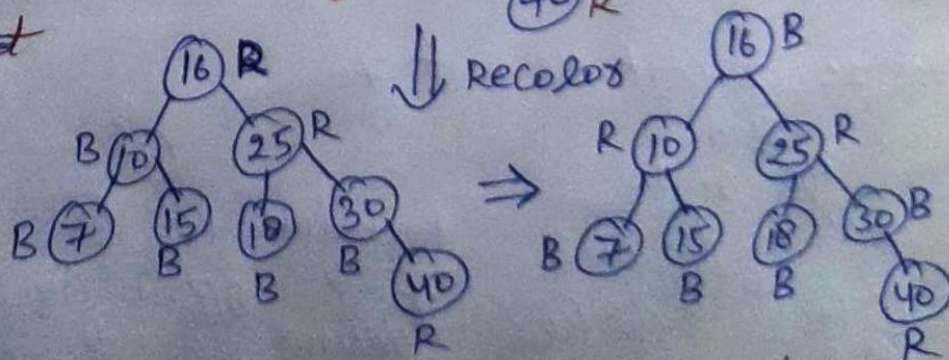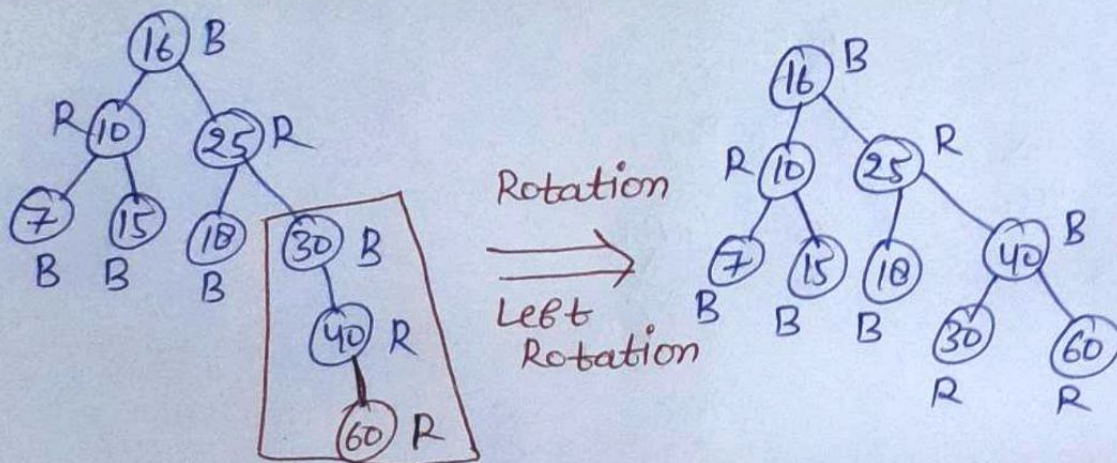

Step 4: Insert element 15

Step 5: Insert element 16

L-R Rotation

Right Rotation

Step 6: Insert element 30

$\Rightarrow$ Recolor

Step 7: Insert element 25

$\Rightarrow$ R-L Rotation $\Rightarrow$ R Left Rotation

Step 8: Insert element 40

Recolor $\Rightarrow$

Red-Red conflict

Left Rotation $\Rightarrow$ Rotation (R-R)

~~Step 9: Insert element~~

$\Downarrow$ Recolor

$\Rightarrow$
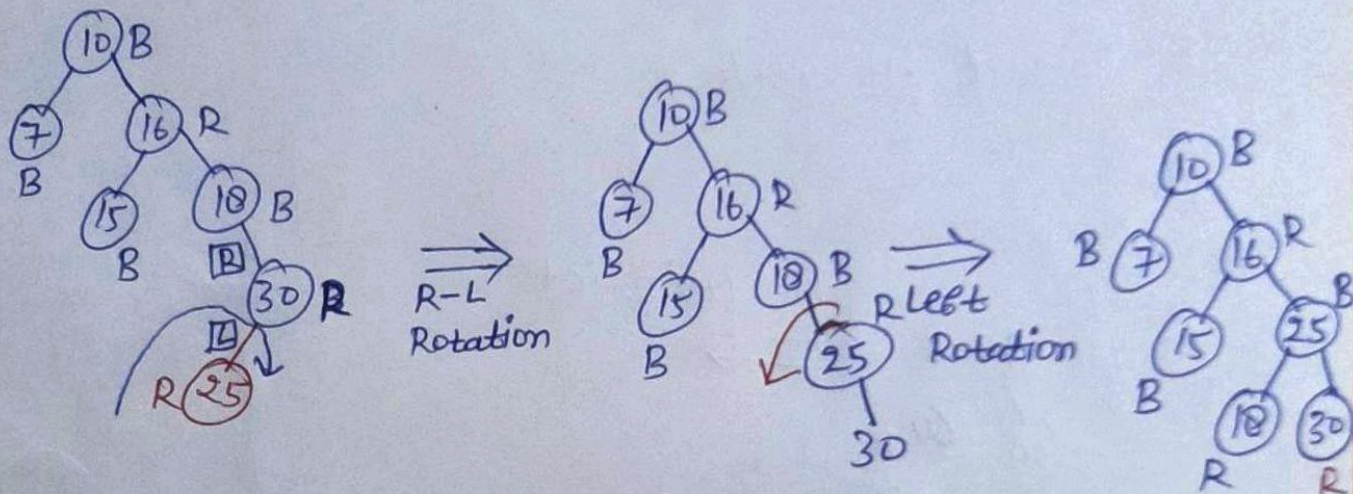
Step 9: Insert element 60

Rotation

Left Rotation

Step 10: Insert element 2

No conflict

Step 11: Insert element 1

Rotation & Recolor

Right Rotation
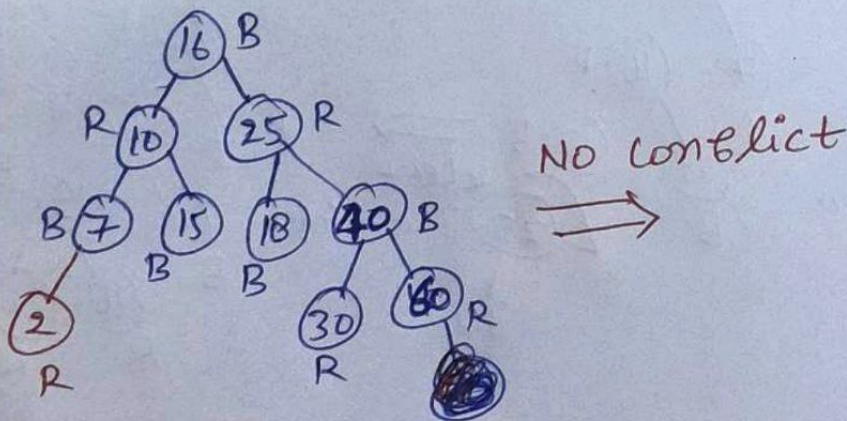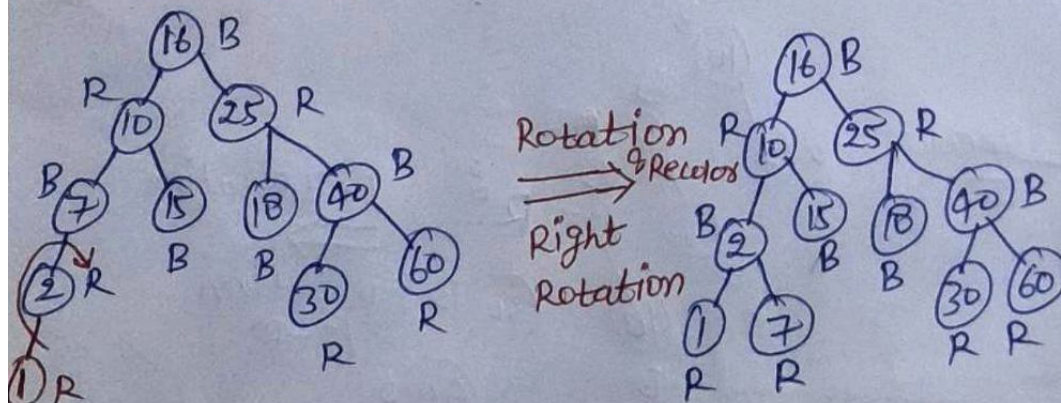
Step 12: Insert element 70

Recolor

Red-Red conflict