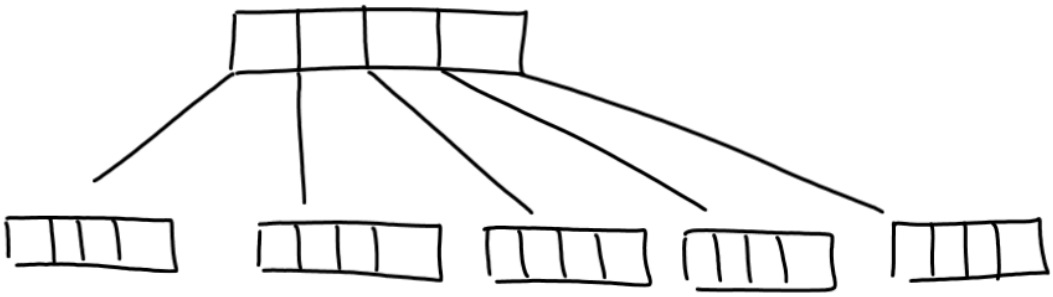
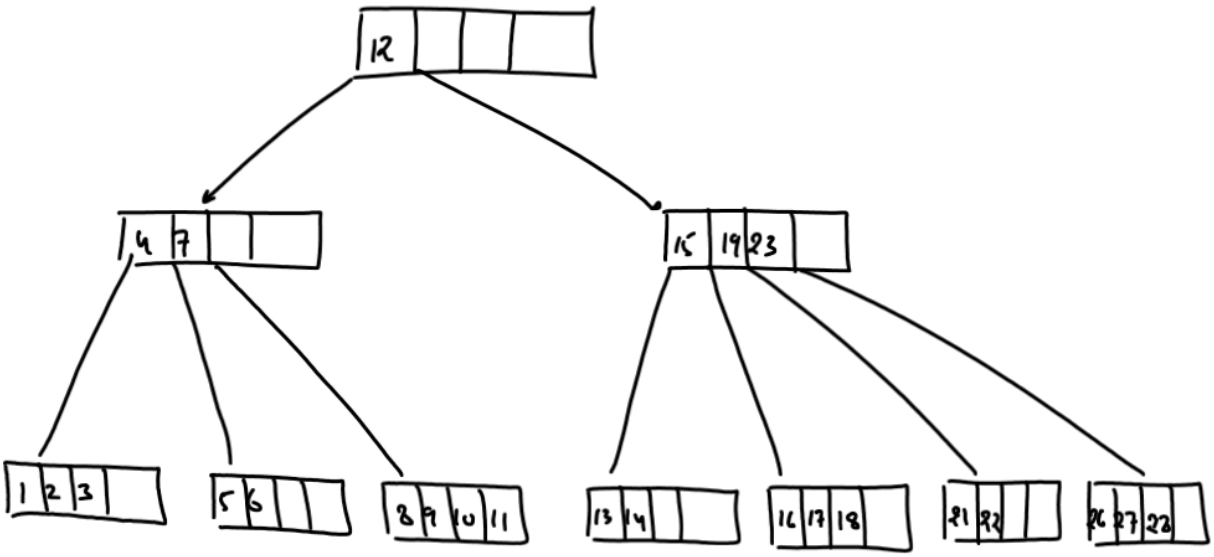
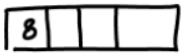
B- Tree



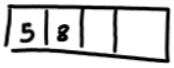


8, 5, 13, 15, 12, 2, 7, 6

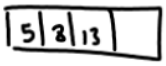
a.



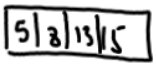
b.



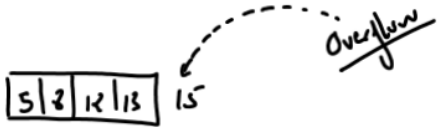
c.

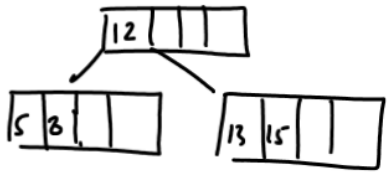


d.



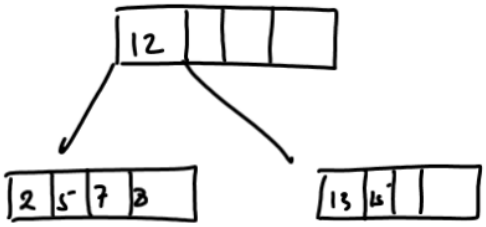
e.

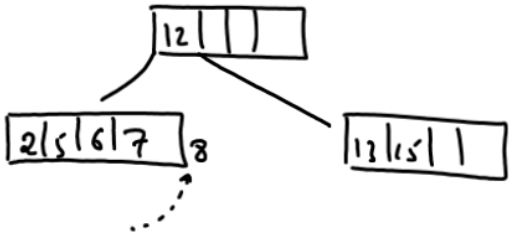


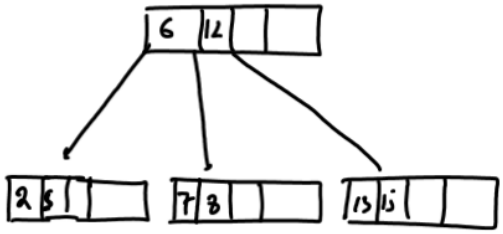


f.

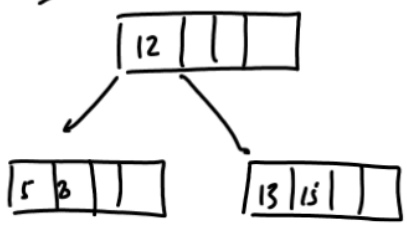
g.



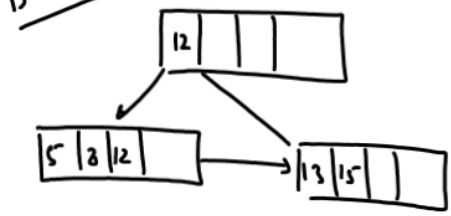




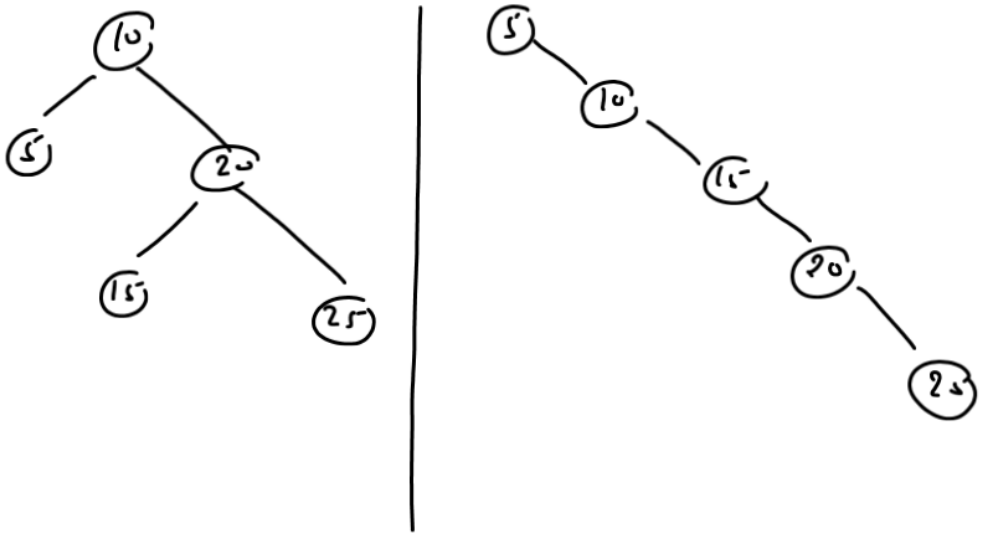
B-Tree



B+ Tree



Red Black Tree



Important Properties Of A Red-Black Tree
==========

1. It is a self-balancing BST

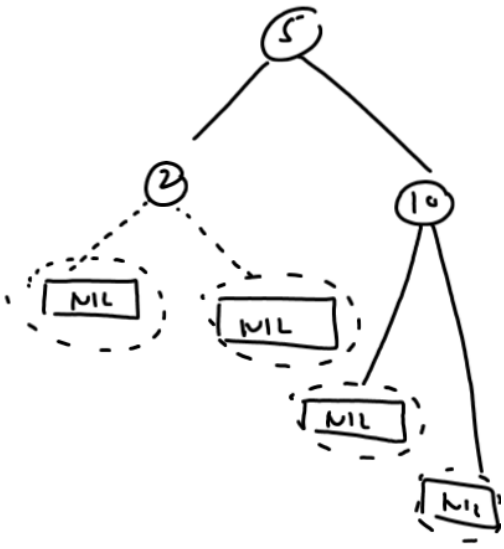
2. Every node in this tree is either RED or BLACK

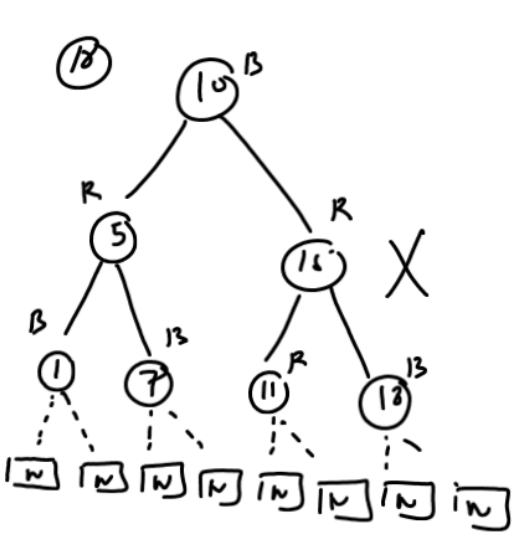
3. Root is always in BLACK color

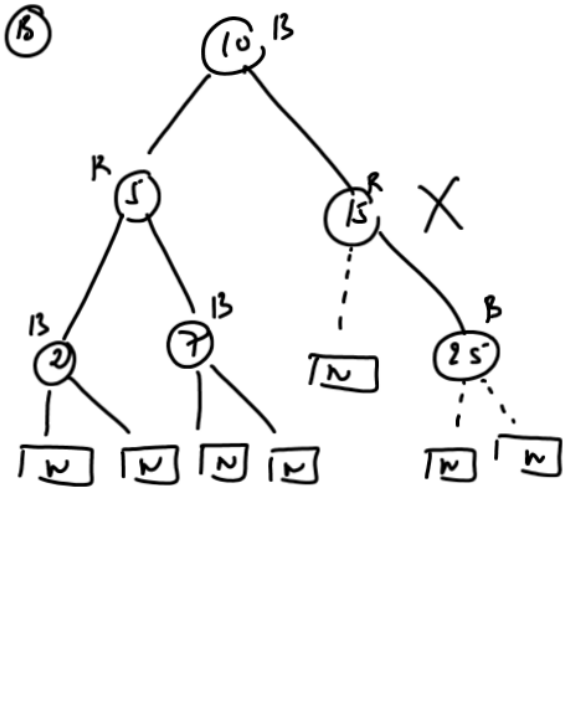
4. Every leaf node which is NIL is always BLACK

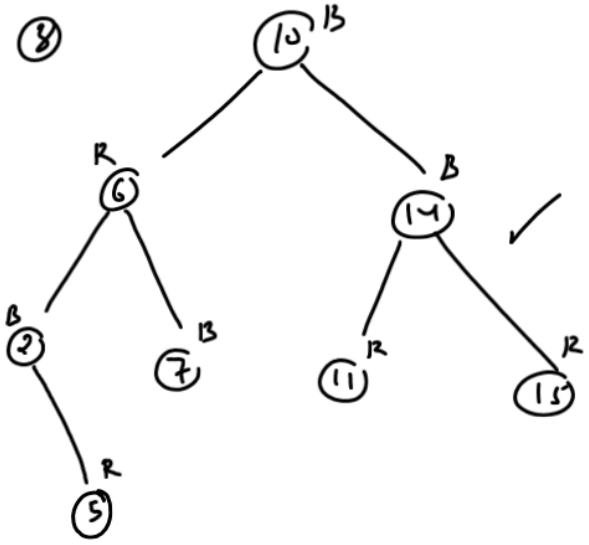
5. No two RED nodes can be parent-child

6. Every path from any node to NIL nodes must always have same number of BLACK nodes









Rules For Inserting a New Node In A Red -Black Tree
==========

1. If the tree is empty , then create a new node as root node with BLACK colour.

2. If tree is not empty , then create a new node as LEAF node and colour it RED.

3. If the parent node of the new node is BLACK then return

4, . If the parent node of the new node is RED then check the colour of PARENT's SIBLING:

a. If the colour of parent's sibling is BLACK or NULL then perform reqd ROTATION and also recolour (revert the colour)

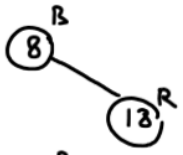
b. If the colour of parent's sibling is RED , then:
1. RECOLOUR both , parent and sibling
2. Check if the parent's parent is not root node , then RECOLOUR it also and also recheck

Constrruct A RED BLACK tree of the followign set of values:

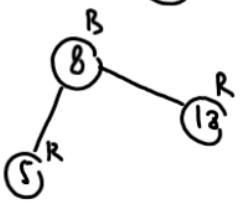
8,18,5,15,17,25,40,80

a.

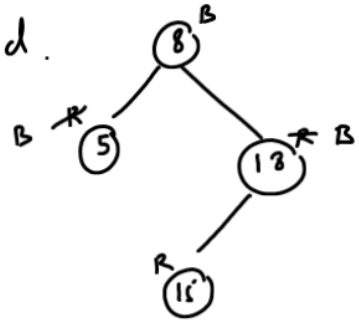
b.



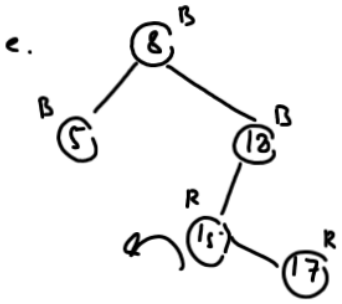
c.



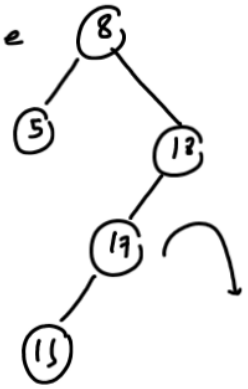
d.



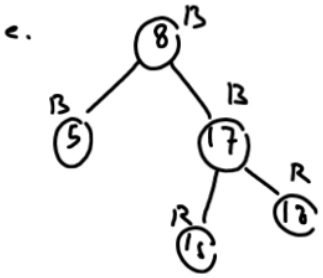
e.



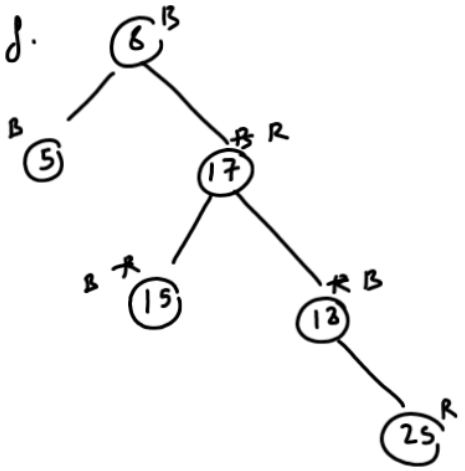
e



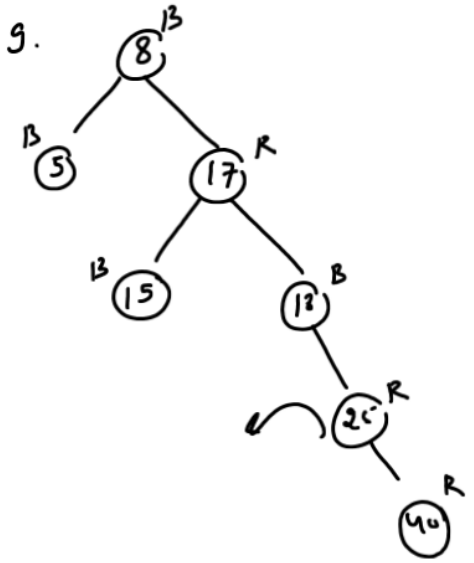
e.



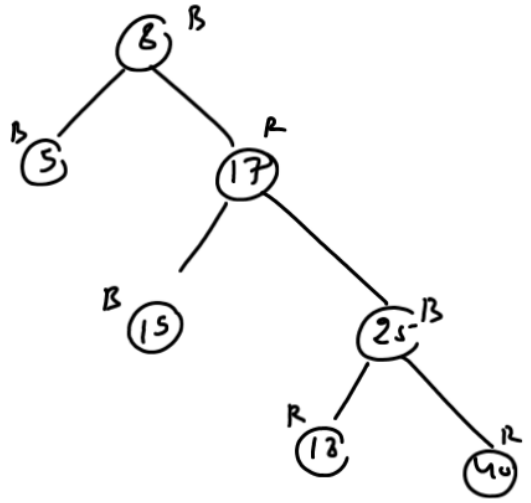
f.



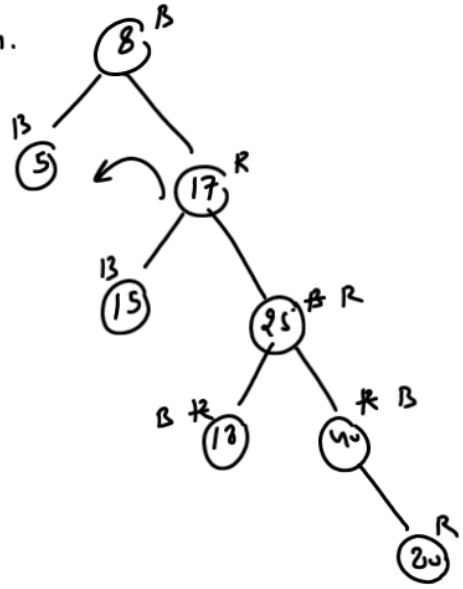
g.

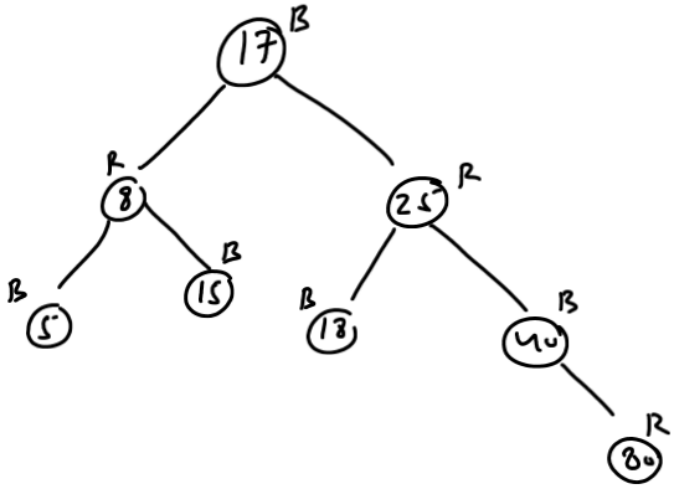


g.


8 B
5 B
17 R
15 B
25 B
18 R
40 R

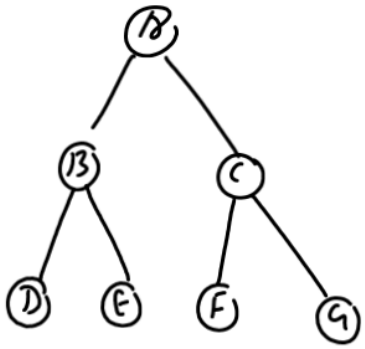
h.


8 B
5 B
17 R
15 B
25 R
18 B
40 B
80 R

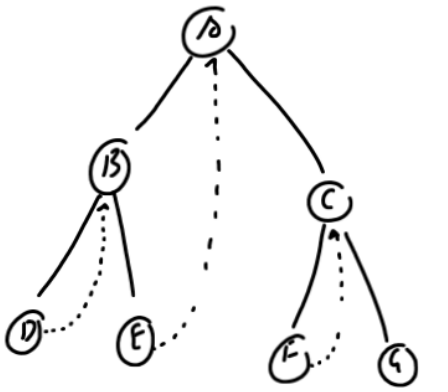

17 B
8 R
25 R
5 B
15 B
18 B
40 B
80 R

Threaded Binary Tree

In: D, B, E, A, F, C, G

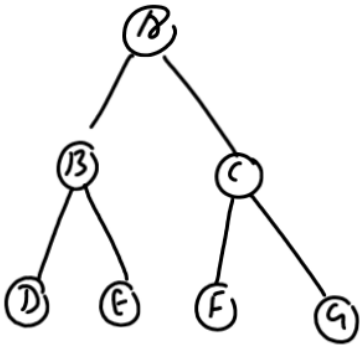


A Normal Binary Tree

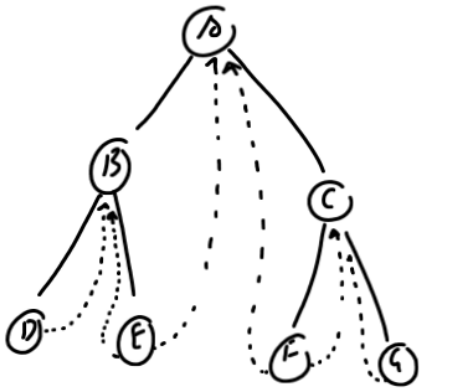


One Way Inorder TBT

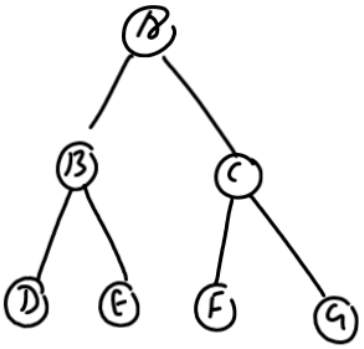
In: D, B, E, A, F, C, G



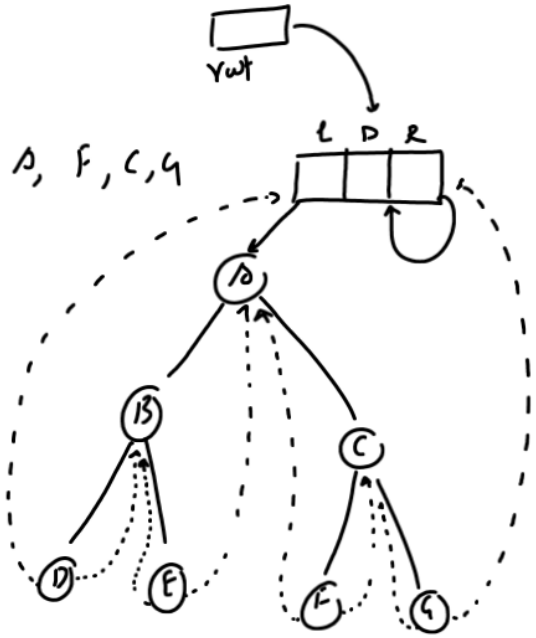
A Normal Binary Tree



Two Way Inorder TBT



A Normal Binary Tree



Two way Inorder TBT with A Header Node