

Name: Shivani Panchiwal
UTA ID: - 1001982478
Assignment: - Hw-2

DAA.

Miracle
Pg.No: _____
Date: / /201

Q.1

(1). what are the least & greatest num of leaf nodes in a binary tree with n nodes. Show ex.?

\Rightarrow Total num. of nodes in full binary tree =

$$2^0 + 2^1 + 2^2 + \dots 2^{h-1}$$

where h is the ~~height~~ height of the tree.

Here, height is 2.

$$\text{So, } n = (1+2)+4$$

where $(1+2)=3$ are num of non-leaf nodes and 4 are the num of leaf nodes.
 n is the total num. of nodes.

Also, if height is 3, $n = (1+2+4) + 8$

So, we observe that n num of leaf nodes is greater than non-leaf nodes in full binary tree.

So, greatest num. of leaf nodes = $\text{floor}[(h+1)/2]$
consider height of tree as 3 therefore,
total num of nodes becomes 15.

So, $(15+1)/2 = 8$ leaf nodes which is correct.

Least num of leaf nodes = Root node can have a single left or right child, which is also interior node, so height will become 1 in this case, & there is exactly one child.
So, least num. of the leaf node is 1.

(2). What's the relationship b/w the sum of nodes in a full binary tree and the sum of leaf nodes, show with an example?

→ Total num of nodes in a full binary tree =
num of non-leaf nodes + no. of leaf nodes.

$$\text{Num of leaf nodes} = \text{Num of non-leaf nodes} + 1$$

$$\text{Num of non-leaf nodes} = \text{Num of leaf nodes} - 1$$

$$\text{Let num of leaf nodes} = n$$

$$\text{So, num of non-leaf nodes} = n - 1$$

$$\text{Total num of nodes} = n + (n - 1) = 2n - 1$$

Hence, num of nodes in a full binary tree = $2n - 1$
where n is num of leaf nodes.

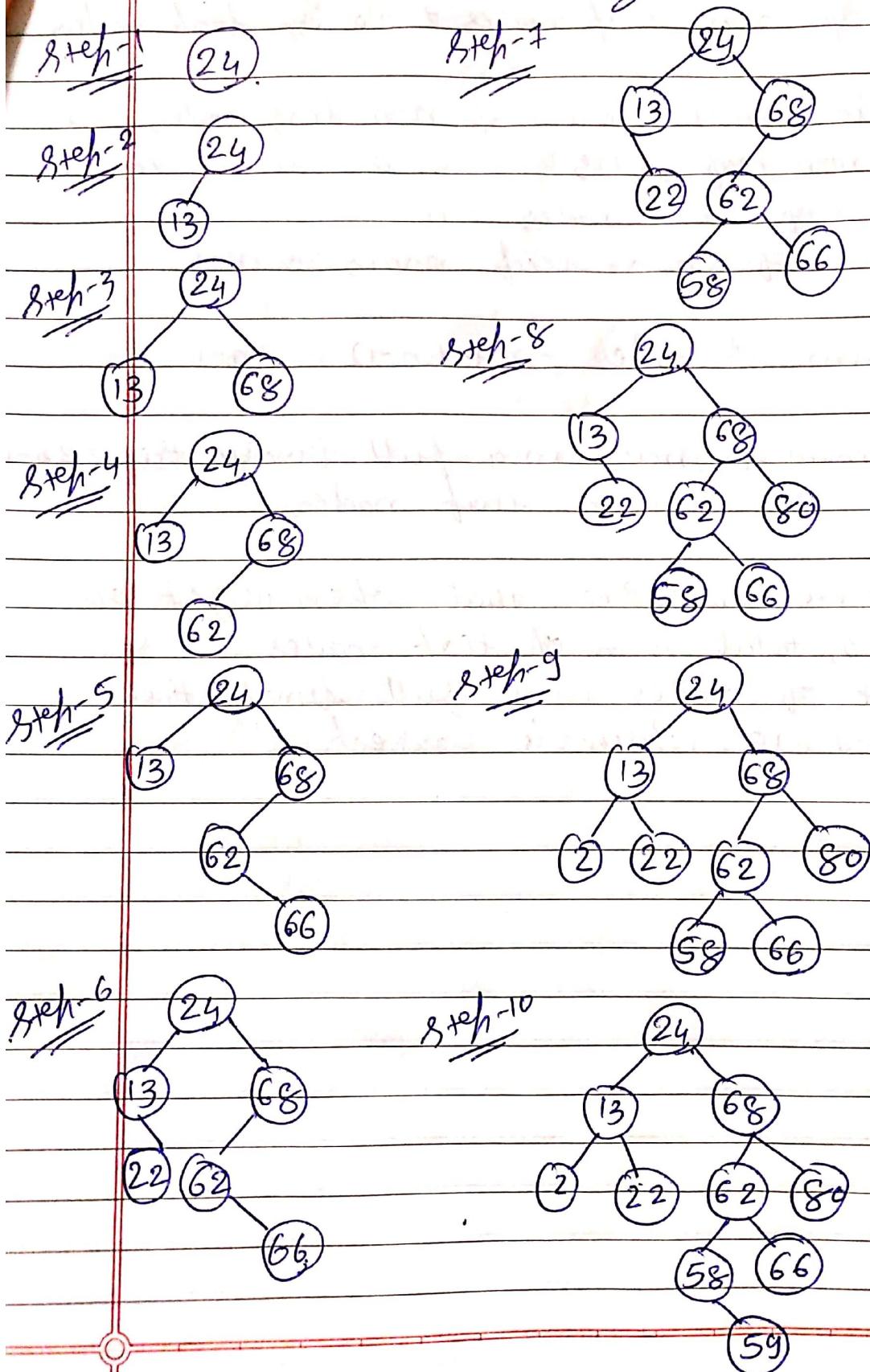
For ex, we have seen that when height of tree is 3, total num of leaf nodes are 8.

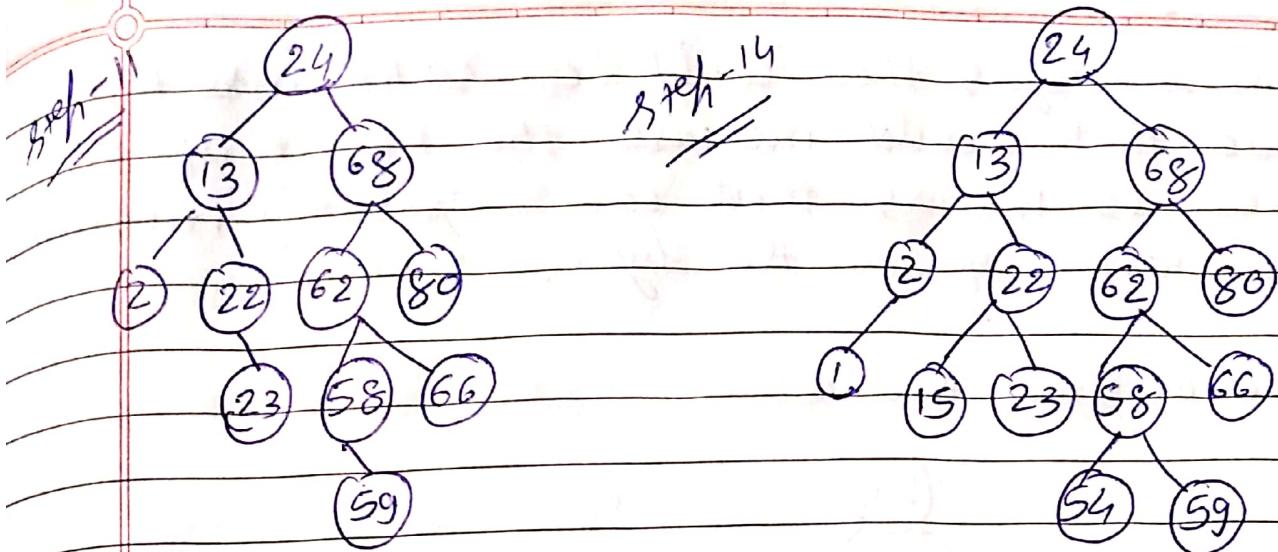
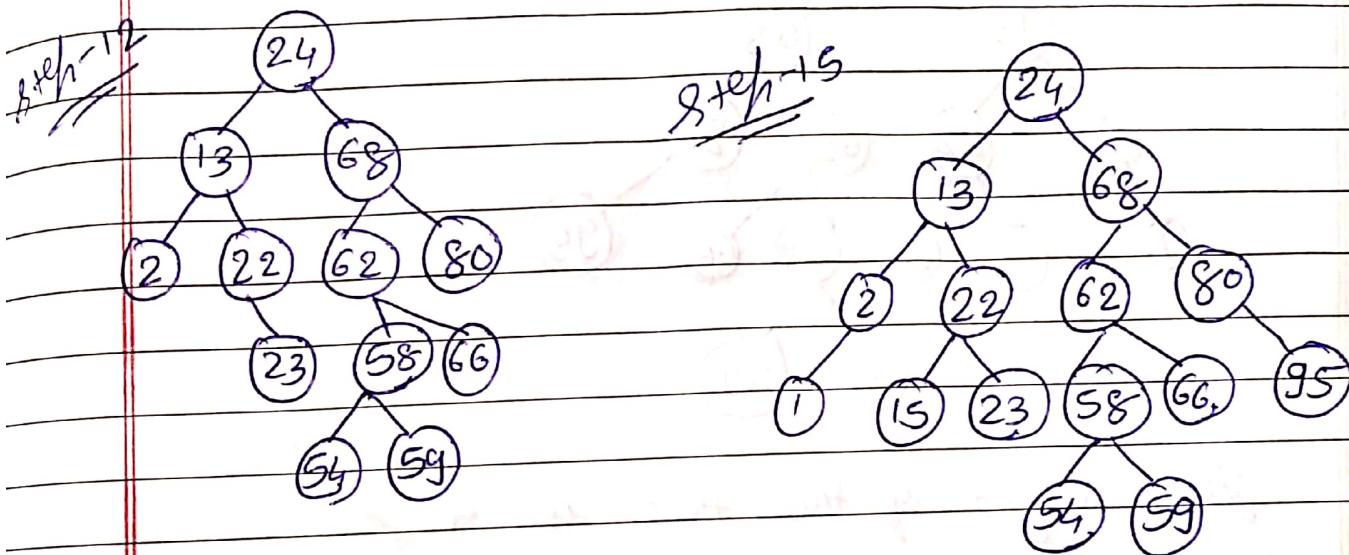
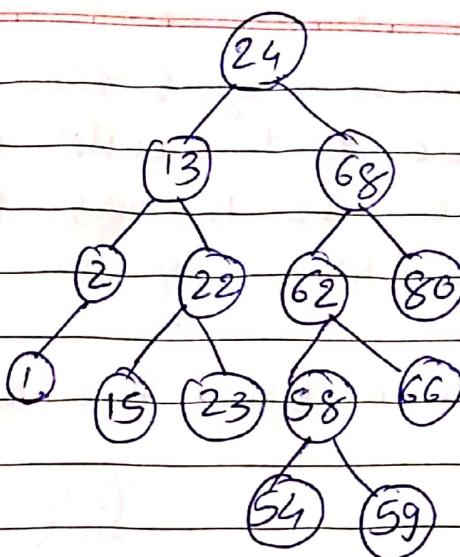
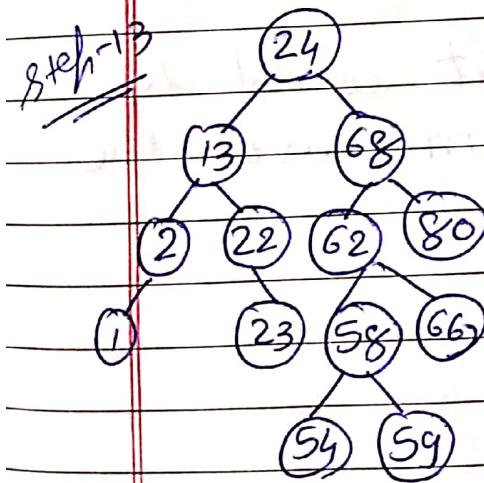
$$\text{So, num of nodes in a full binary tree} \\ = 2 \times 8 - 1 = 15 \text{ which is correct.}$$

Q.2

- (1) 24, 13, 68, 62, 66, 22, 58, 80, 2, 59, 23, 54, 1, 15, 95

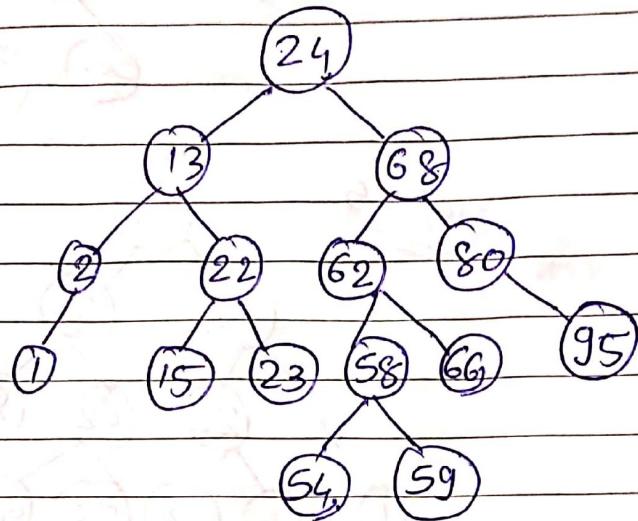
⇒ Below are the steps in which the obj. will get inserted into the binary search tree.



~~8xeph-14~~~~8xeph-15~~

(2). Give 5 integers that could be inserted into this tree that would increase the height of this tree. (5 integers that can separately increase the height of the tree by 1). Explain.

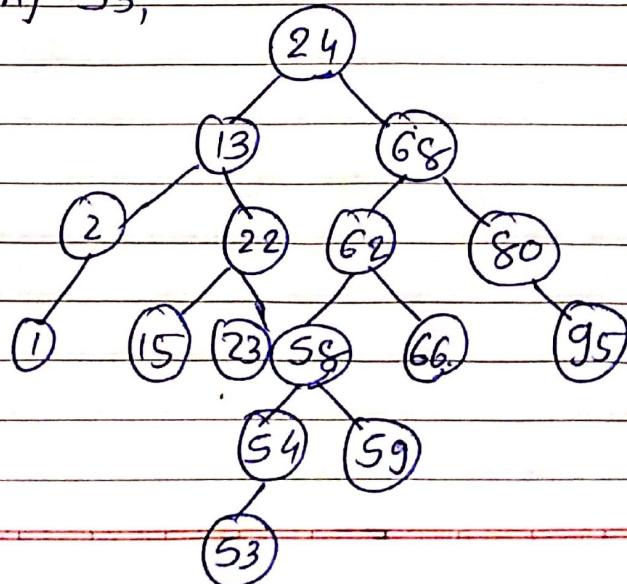
⇒ The original tree is :-



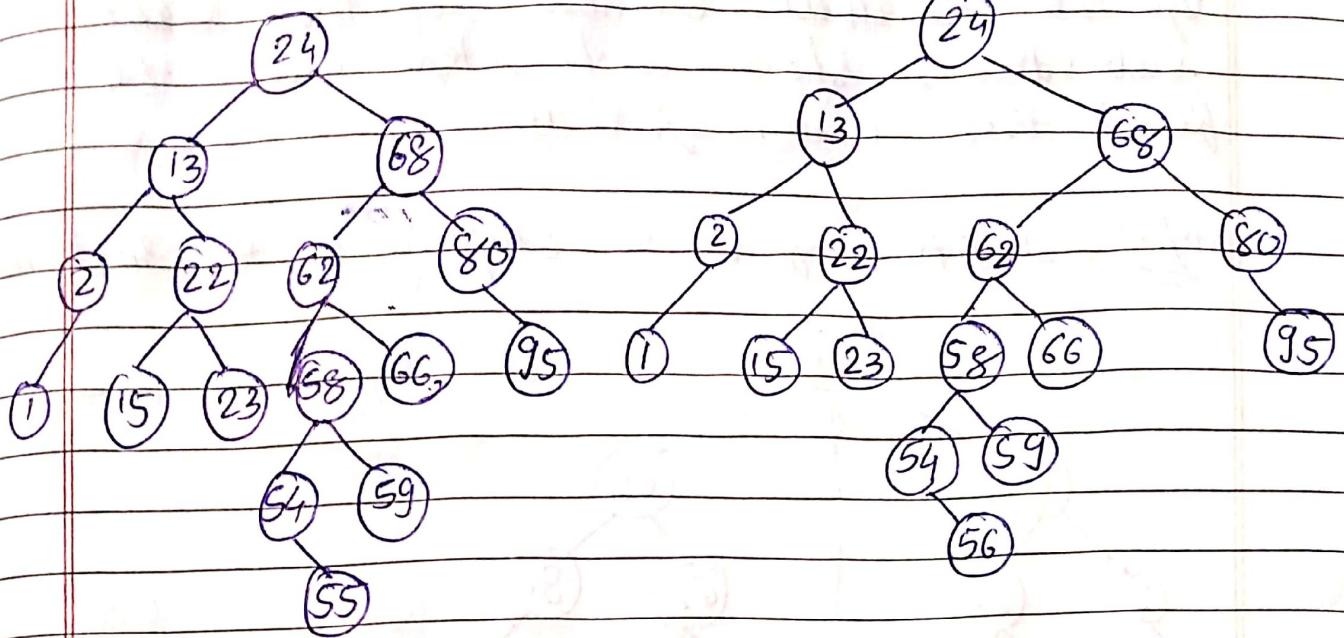
The height of the tree is 4.

⇒ The 5 elements (integers) that could be inserted into this tree to increase the height by 1 are,
53, 55, 56, 60, 61

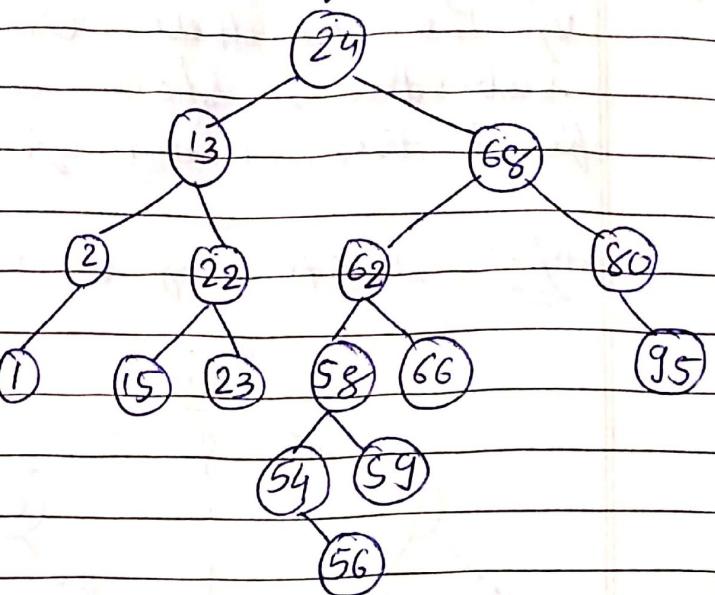
By Adding 53,



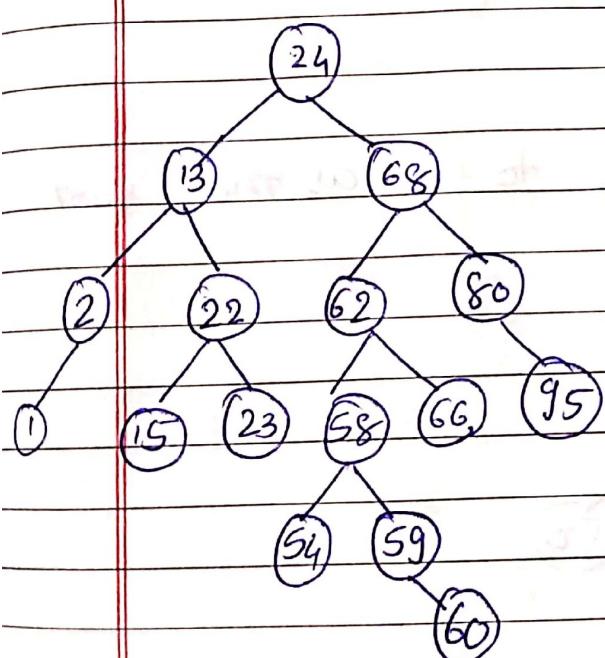
By Adding 55,



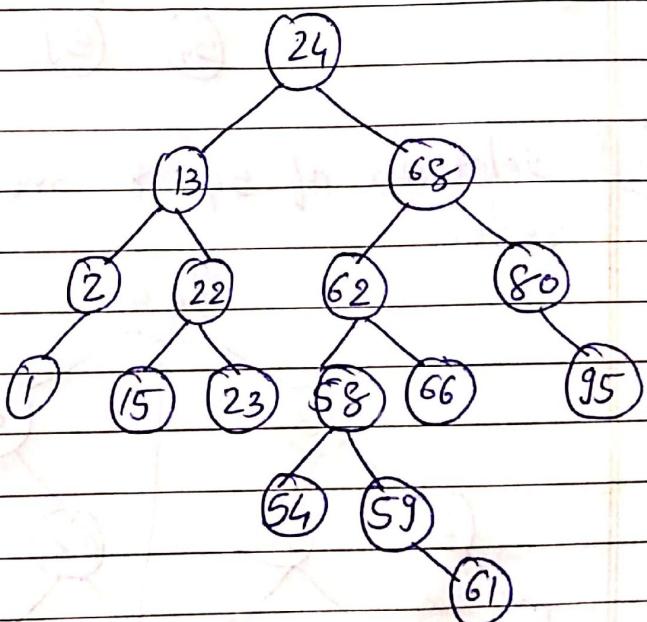
By Adding 56



By Adding 60,



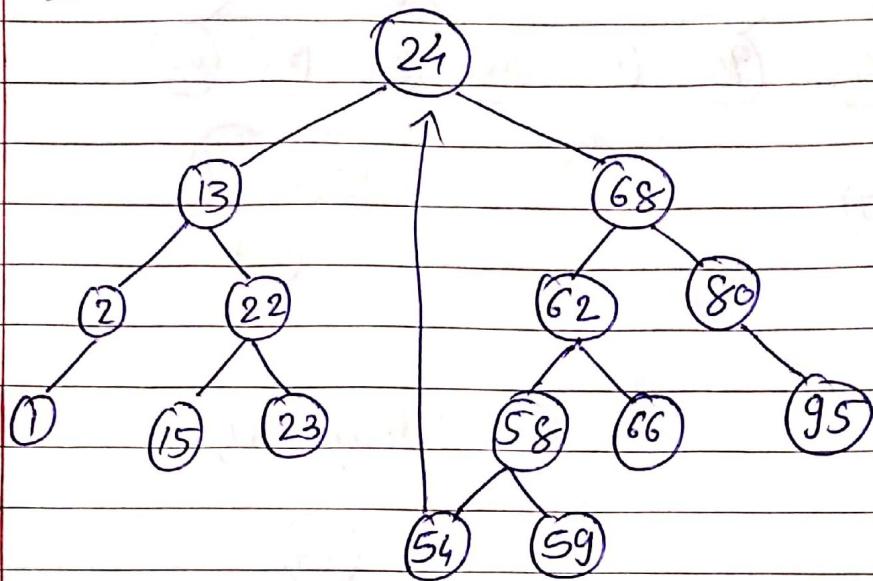
By Adding 61,



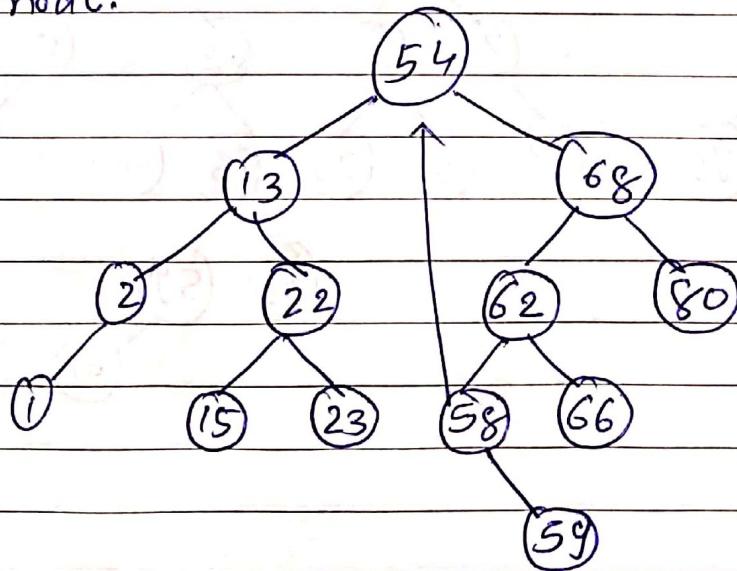
(3).

Remove the root node four times by copying up the smallest element of the eight sub-tree, show each step and the final tree. (from fact A).

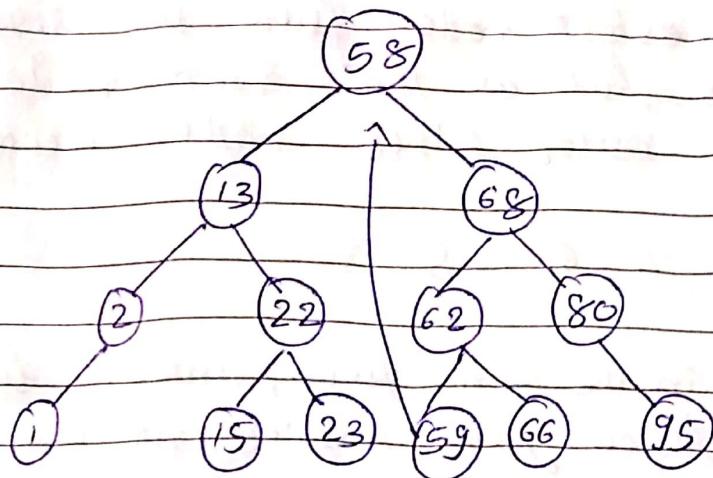
\Rightarrow Step-1 Deletion of 24 & shifting 54 to root nod



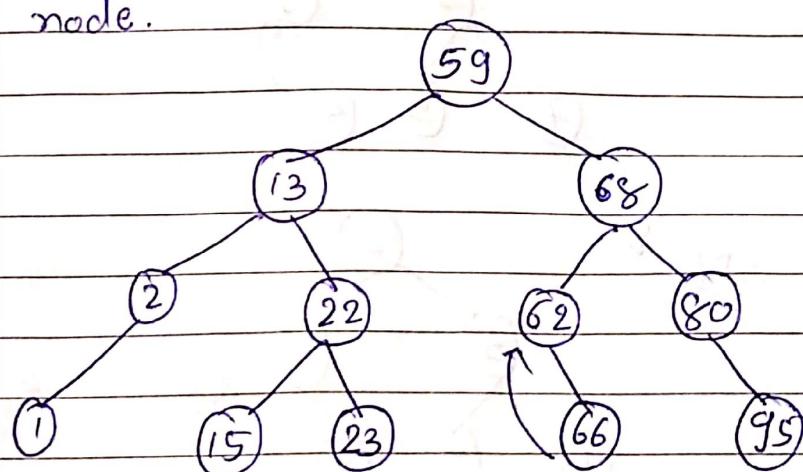
~~Step-2~~ Deletion of 54 & moving to 58 as the root node.



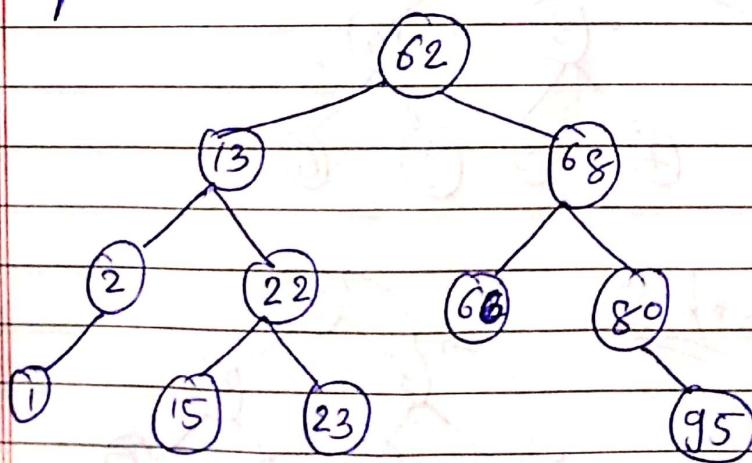
~~Step 3~~ deletion of 58, & insertion of 59 as root node.



~~Step 4~~ deletion of 59, and insertion of 62 as the root node.



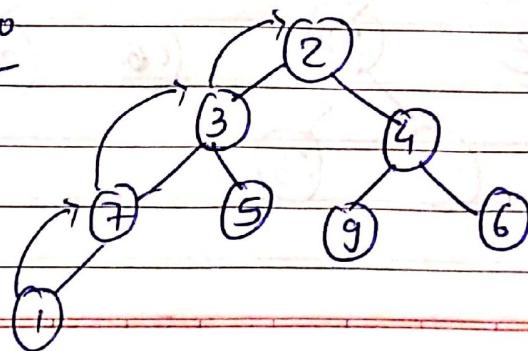
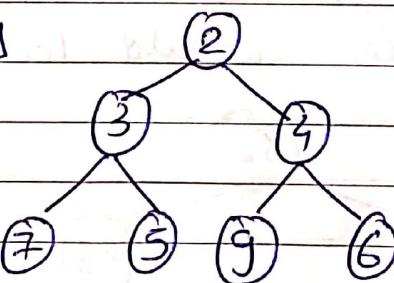
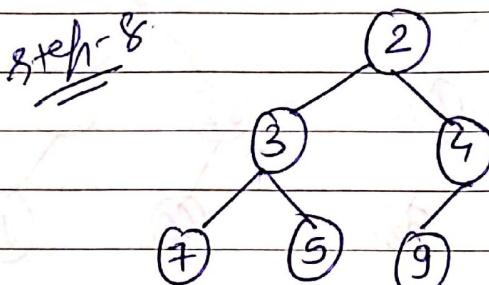
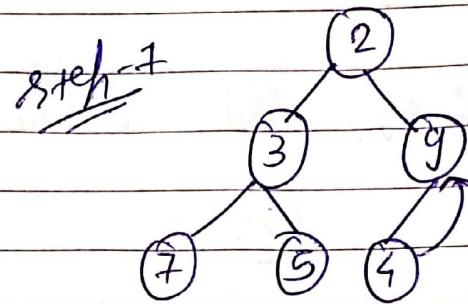
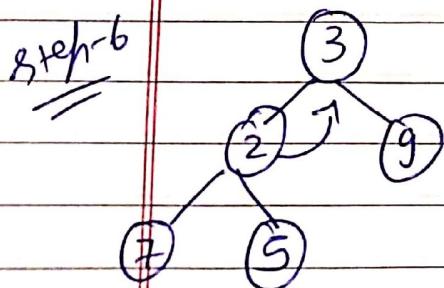
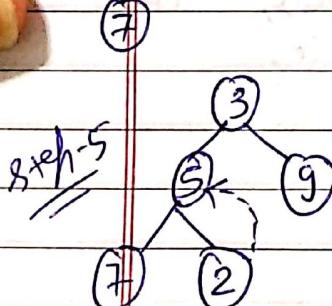
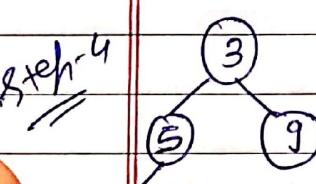
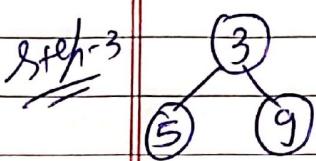
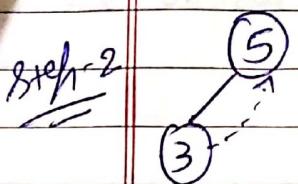
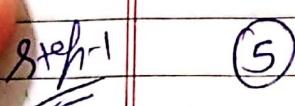
~~Step 5~~ After deleting the root node 4 times, the final tree would look like this.

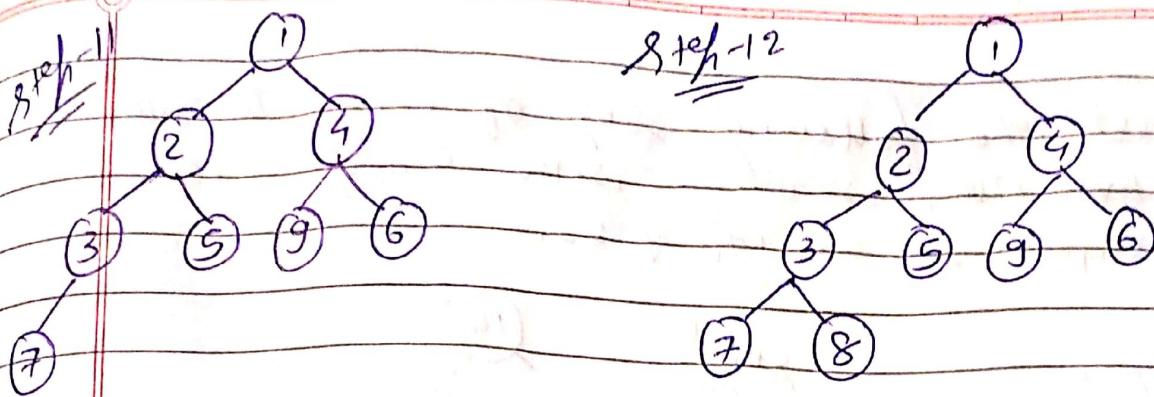


⇒ In each step, the root node was removed and the shortest node from the right subtree was put as the root node arranging the other nodes accordingly.

Q.3 5, 8, 3, 9, 7, 2, 4, 6, 1, 8

insert objects in the order given, into a binary min-heap & place your answer into an array.





final Ans:-

1	2	4	3	5	9	6	7	8
---	---	---	---	---	---	---	---	---

⇒ In Step 2, node 3 is less than node 5 so, we have swapped the nodes.

In Step 5, when 2 is inserted into the tree, the definition of min-heap gets violated and so, we swapped 5 with 2.

In Step 6, we swapped the root node 3 with 2.

In Step 7, again after adding 4, we swapped it with 9 as 4 is lesser than 9.

In Step 10, after adding 1, again min-heap violation occurs so, we swap 1 to the root node of the tree.

Q.4

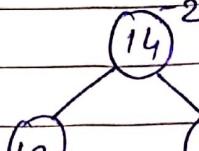
- (1). Insert the following seq. of elements into an AVL tree, starting with empty tree.

12, 24, 14, 27, 35, 17, 19, 22.

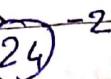
Step-1



Step-6

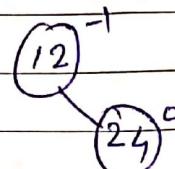


-2



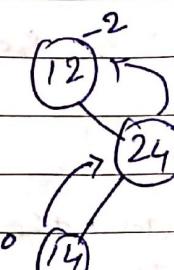
-2

Step-2

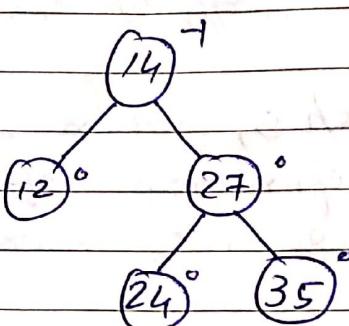


-1

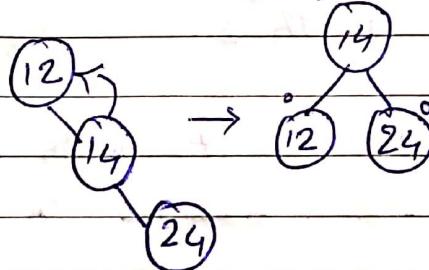
Step-3



Step-7 LL Rotation.

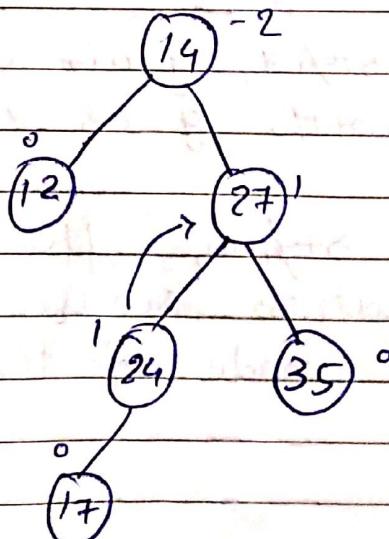


Step-4

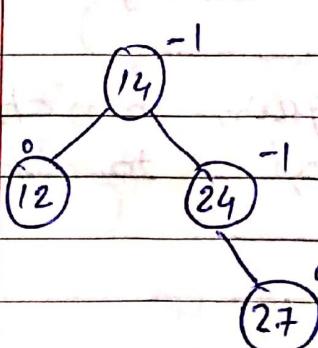


RL Rotation.

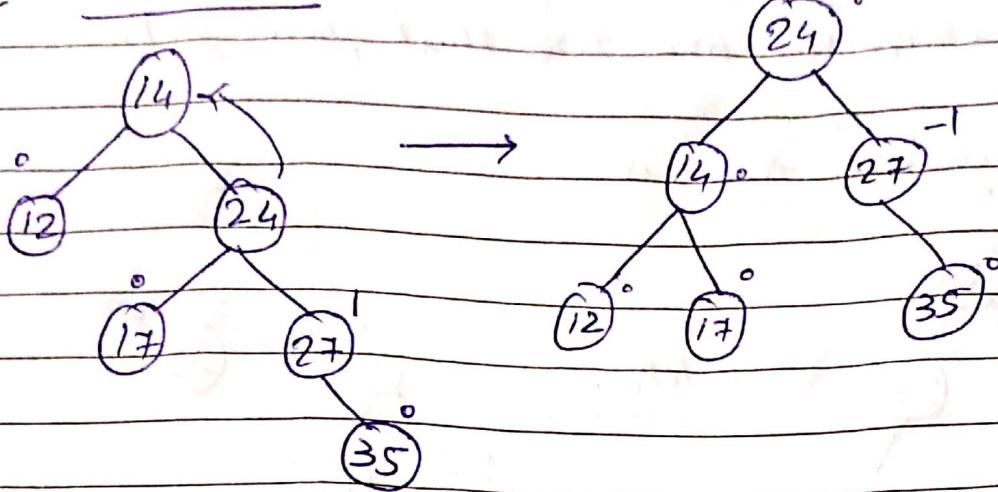
Step-8



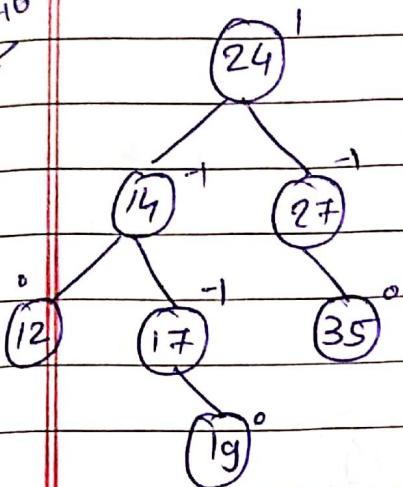
Step-5



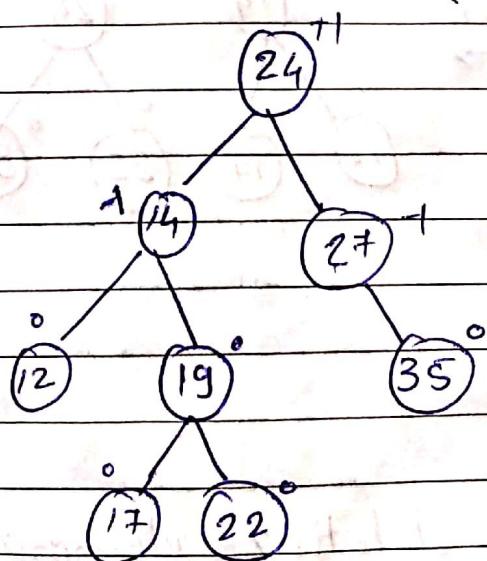
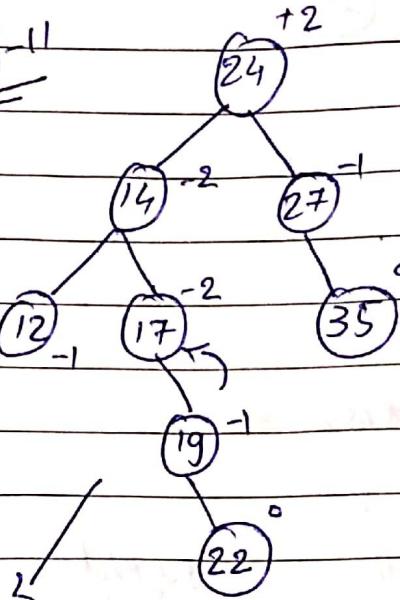
~~Step 9~~ RL Rotation.



~~Step 10~~



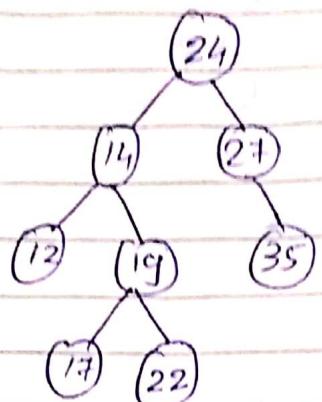
~~Step 11~~



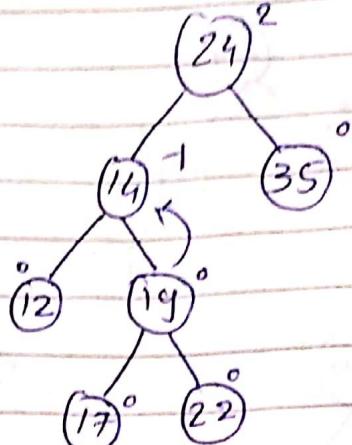
→ The above tree as a result of ~~Step-11~~ is the balanced AVL tree.

(2). delete 27 in the AVL tree that you got.

→ The original tree is :-

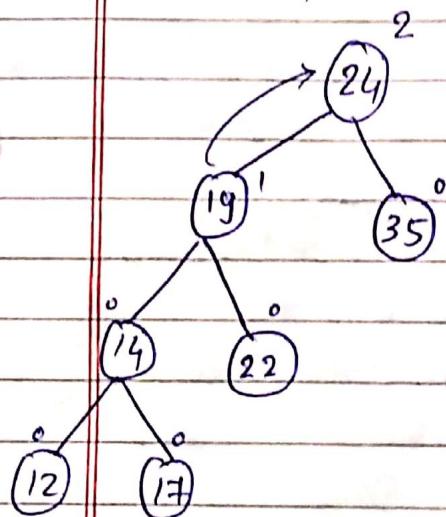


Delete 27 →

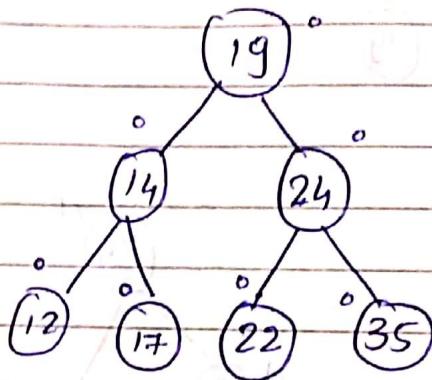


LR Rotation.

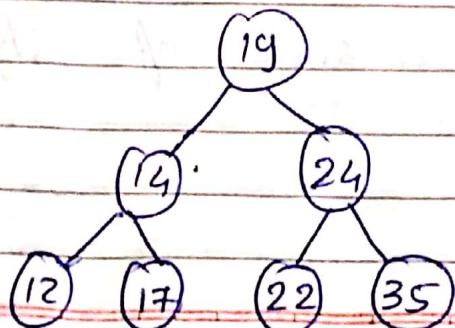
Left Rotation →



Right Rotation →



⇒ final AVL tree after deleting element 27 :-



(3). what maximum difference in height between the leaf of a AVL tree is possible?

→ The condition of AVL tree for each node in BST always have a balance factor of -1 or 0 or 1. Thus, maximum difference in heights between the leafs of a AVL tree is 1.

Q5

(1). What are the operations that could be performed in $O(\log n)$ time complexity by red-black tree?

→ The operations that could be performed in $O(\log n)$ time complexity by red-black tree.

Insertion:- Inserting an element into the Red-black tree.

Deletion:- Deleting an element from the Red-black tree.

Search:- Searching for a particular elements or its predecessor (or) successors in the Red-Black tree.

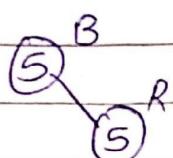
(2). Insert the following sequence into a red-black tree.

5, 6, 1, 9, 2, 4, 3, 8, 7.

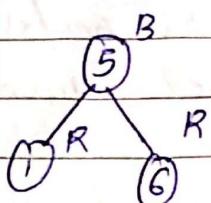
\Rightarrow Step-1 Root Node:-

$$5^R \rightarrow 5^B$$

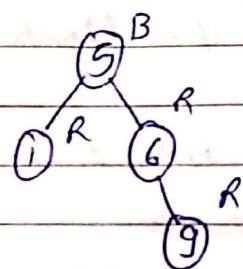
Step-2



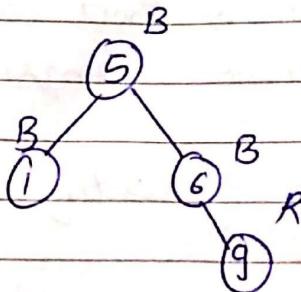
Step-3



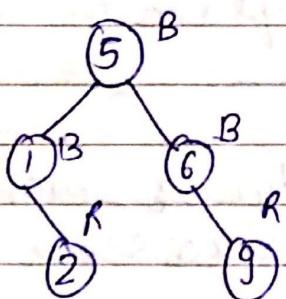
Step-4



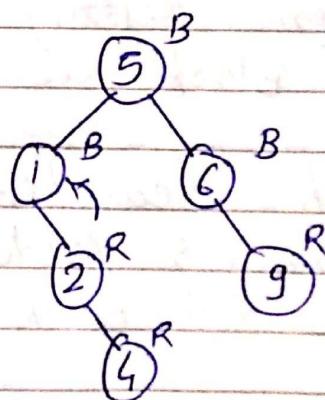
Recoloring



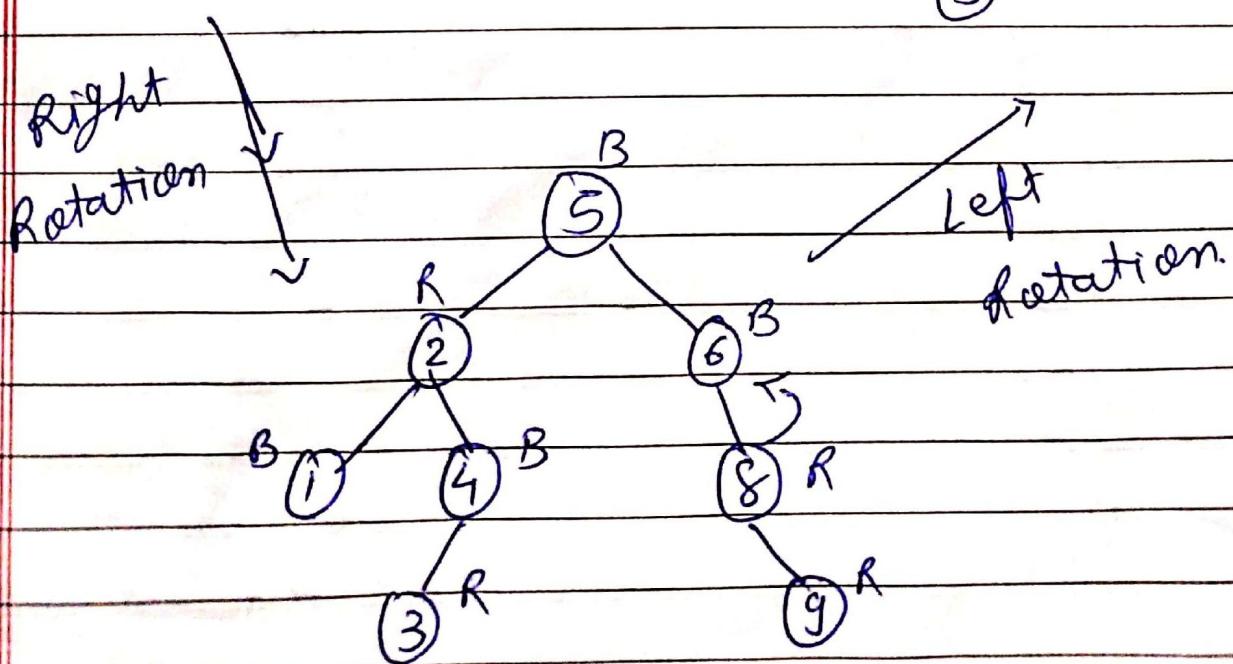
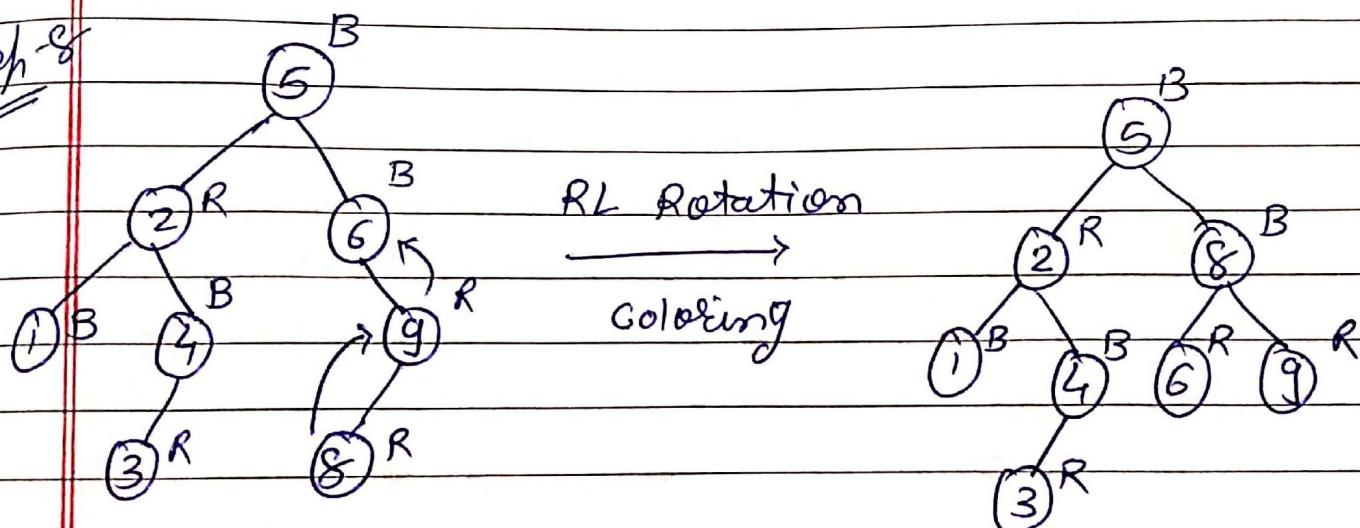
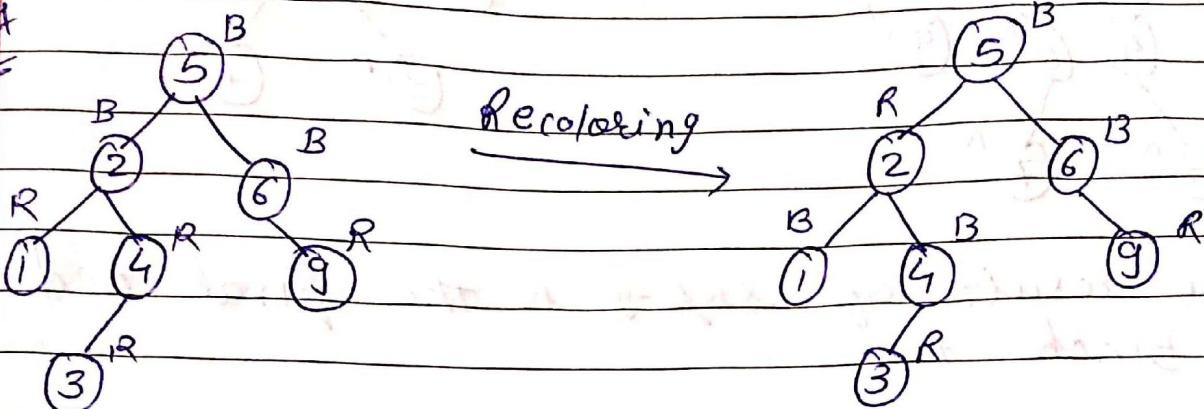
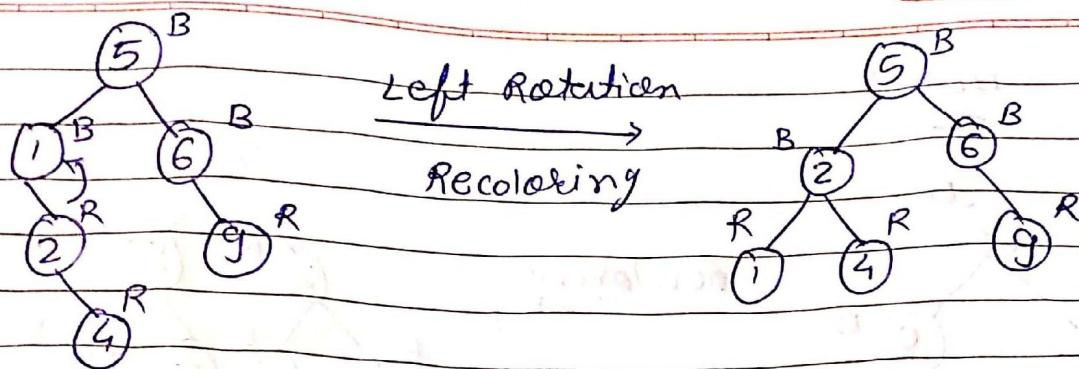
Step-5



Step-6



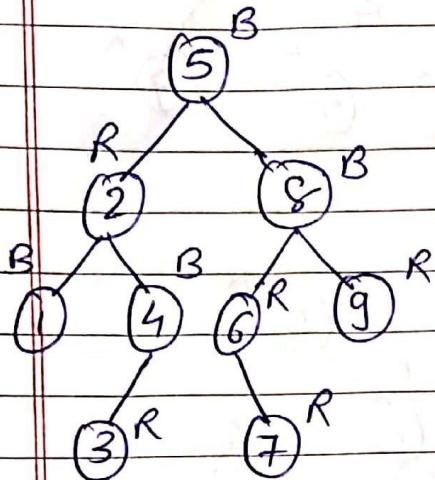
\rightarrow Applying left rotation & recoloring



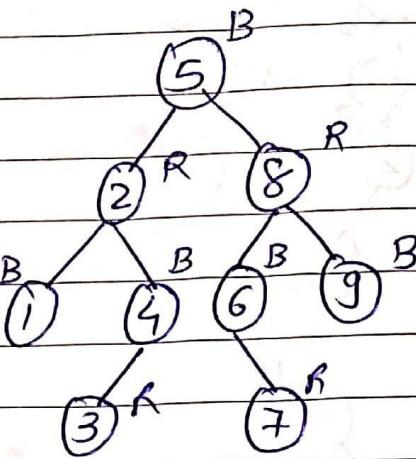
~~Step - g~~

Add 7.

=



Recoloring



→ The result of step - g is the final Red Black tree.