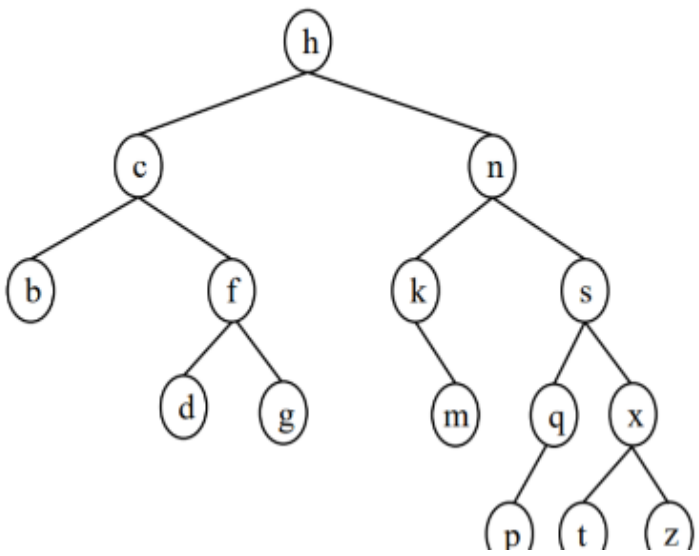
Question:

Given is the BST topology below. The letters are the keys of the BST considering their alphabetic order

a) Check whether it is an AVL tree, showing the subtree heights for each node and clearly state your conclusion! If any violation, state the α node and the type of violation and restore the AVL property!

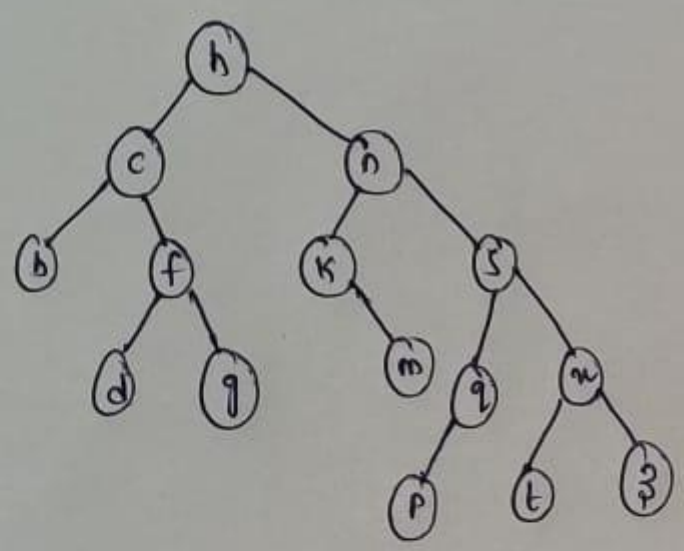
b) Insert the key ***y*** into the BST! Test for the new tree's AVL property by
   i. showing the nodes to be checked in the order from the first to the last and, for each, its subtrees' heights;
   ii. if any violation, stating the α node, the type of violation and the way to restore it;
   iii. and drawing the topology with the AVL property restored using the letters.

c) Insert the key ***e*** into the new AVL tree from part(b)! Test for the new tree's AVL property by going through the same procedure as in part (b) explained in (i)-(iii).



Answer:

Given tree



AVL tree :

1) It must be a Binary search Tree.

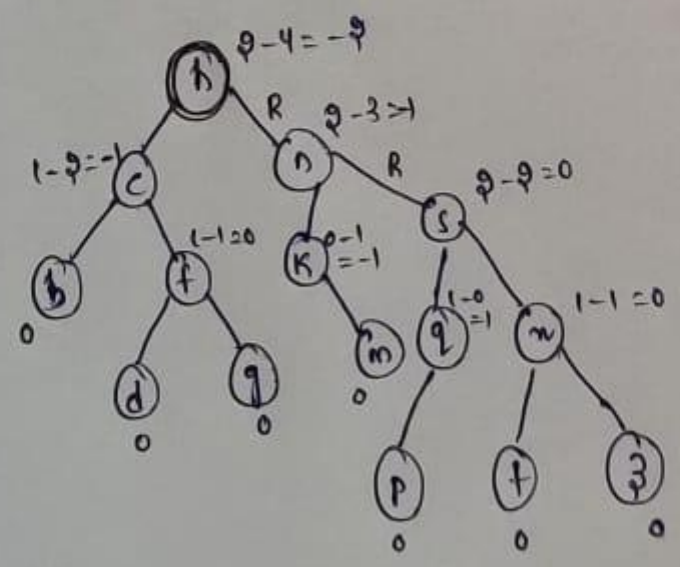
2) It balance factor (BF) is in {-1, 0, 1} values only

BF = HLST − HRST • for all leaves BF = 0.

HLST → Height of Left sub Tree

HRST → Height of Right sub Tree

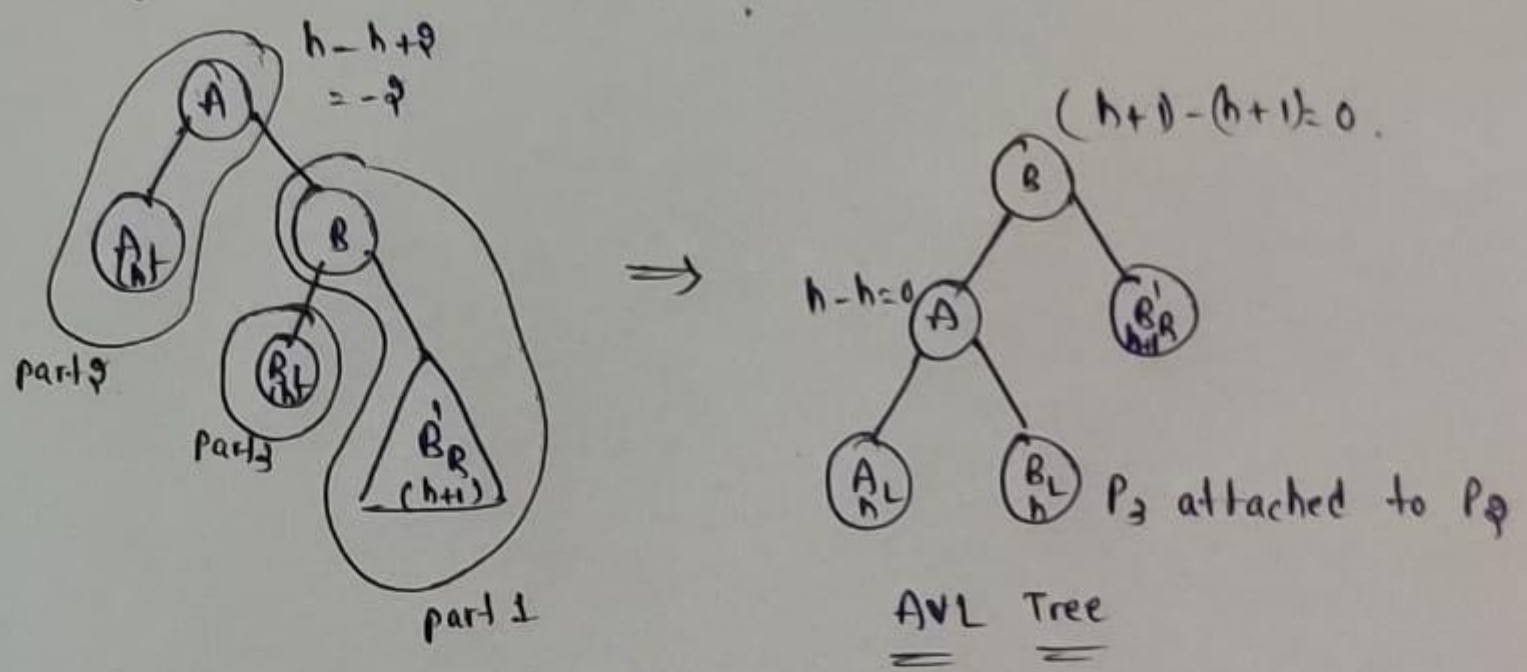
a) so for the given BST BF is as follows



Given tree is not AVL

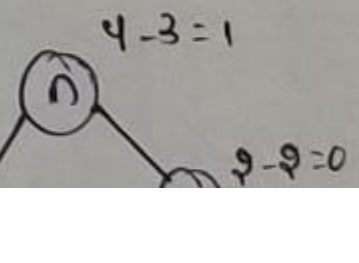
At 'h' node the BF value is −5 which violates the AVL Tree properties it is RR imbalance

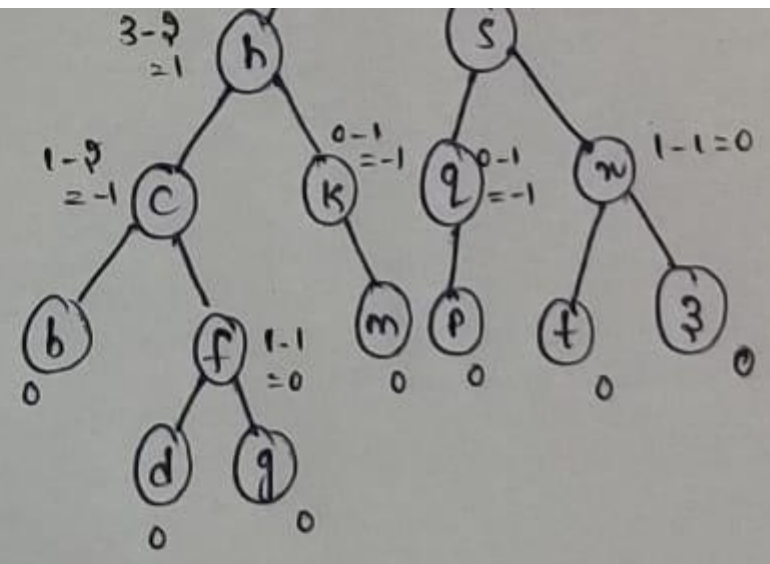
Restoring as AVL Tree



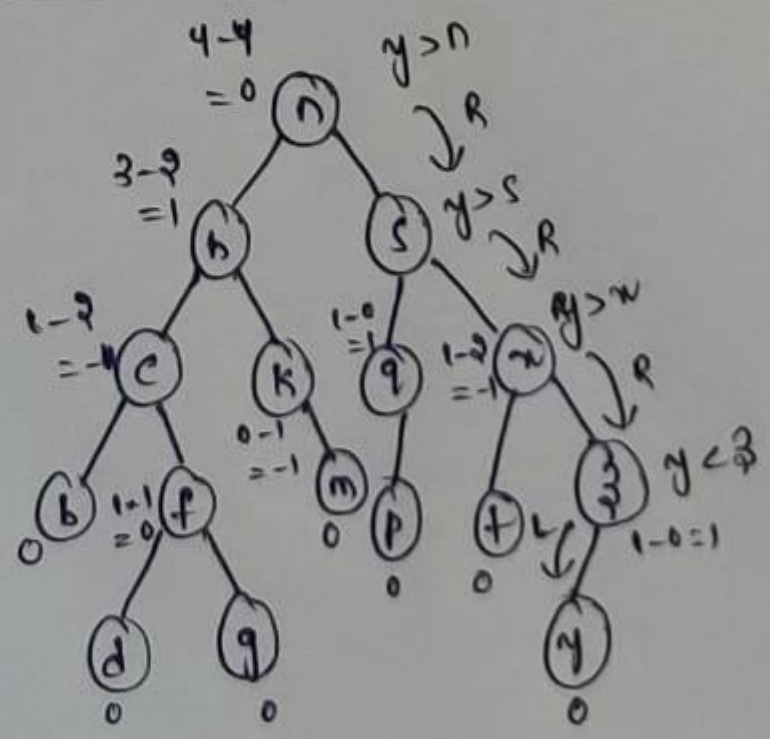
AVL Tree

given:



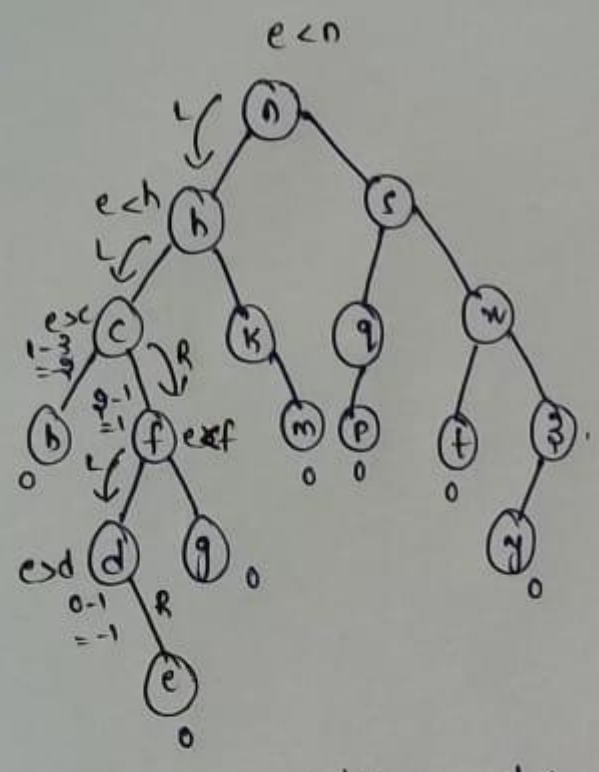


b) Inserting y :



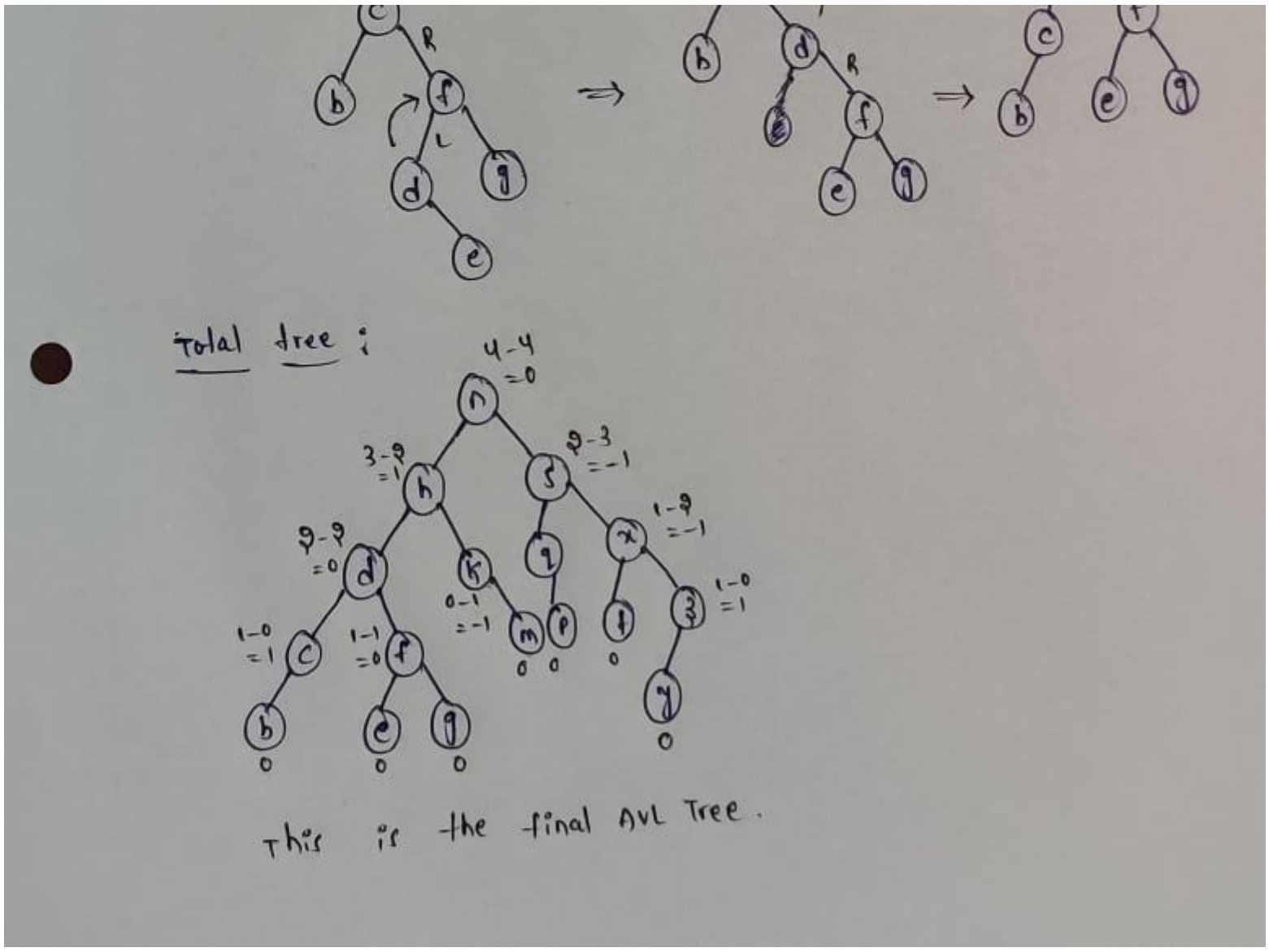
- By inserting 'y' into the tree, still the BST is AVL Tree itself.
- So there is no violation occurs. so the final tree is beside tree only.

c) insert e :



- So at 'c' violation occured and the violation is RL violation
- To remove RL violation first we have to perform LL and then RR.

Taking only violation part :


R
L
R
Total tree :
4-4
=0
3-9
=1
9-3
=-1
9-9
=0
1-9
=-1
1-0
=1
1-1
=0
0-1
=-1
1-0
=1
This is the final AVL Tree.