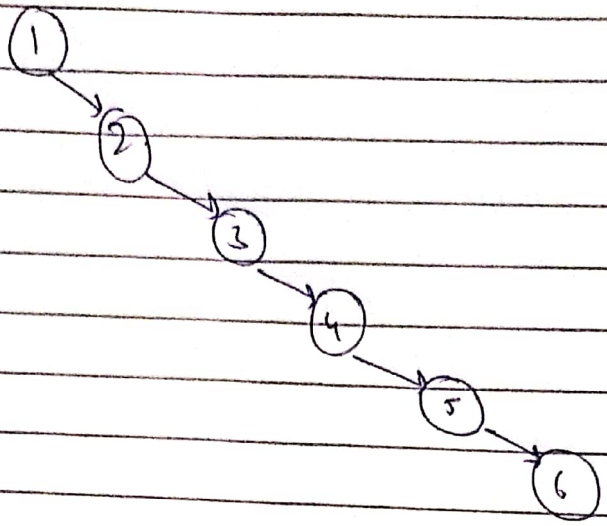
20/4/20 DS-LAB

(1) Created a binary search tree using the given data.




then for each node, calculate Balance factor using formula

B.F. = height(L.S.T.) - height(R.S.T.)

↓

if {-1, 0, 1} → Balanced

else → Unbalanced

so stored these unbalanced nodes in an array Unbalanced_nodes[ ] using counter k

Printed these unbalanced nodes : 1 2 3 4

Then height function was created using,

height(t) = 1 + max(height(t→right), height(t→left));

Here, height for ~~the~~ leaf nodes was taken 1

So height of tree was height(Root) - 1.

(2) Took an array input[] = {1,2,3,4,5,6}

Here each element was taken, and inserted in the tree and simultaneously, the tree was checked for Balanced property.

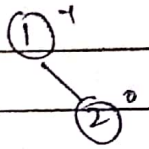
If tree remains balanced, keep inserting.

else, make the tree balanced and then resume inserting.

Step 1.

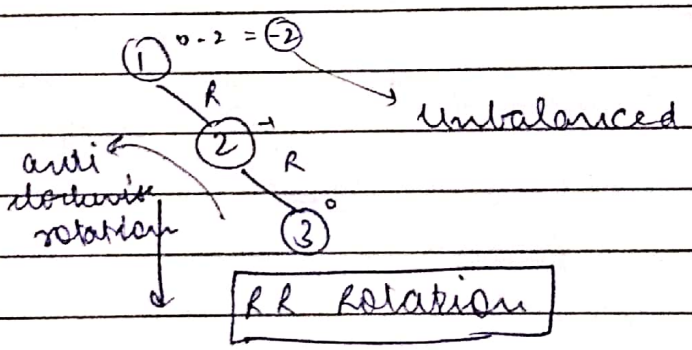

Balanced

Step 2.

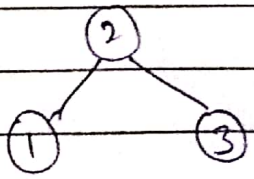



Balanced

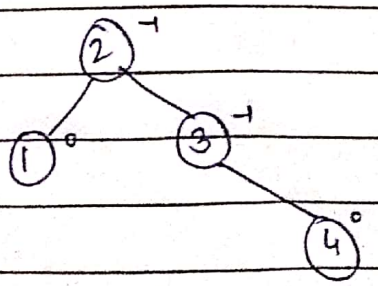
Step 3.




Step 4.

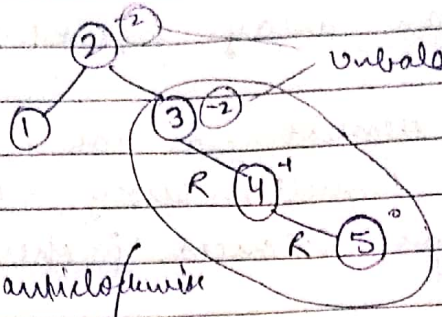



Step 5

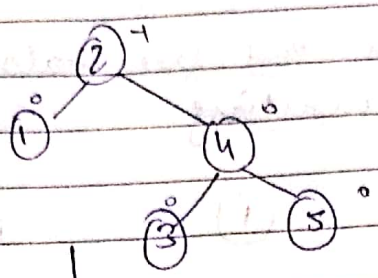



Balanced

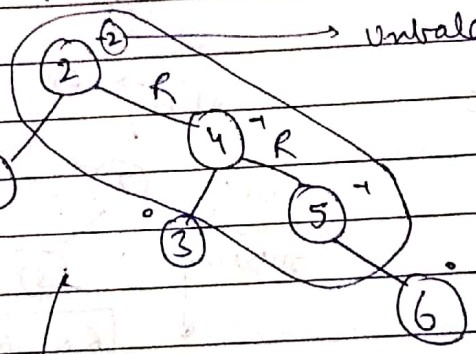
Step 6




Step 7

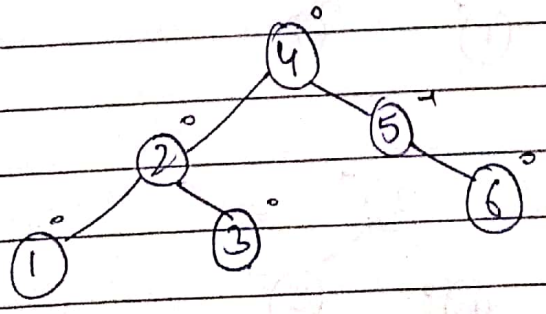



Step 8




drop 3
and anticlockwise rotation

Step 9




so this will be the final tree
with height = 3
and rotations will be:

| |
|---|
| RR |
| RR |
| RR |

Respected Sir,

The first program is running well
But there is some error in the
second code.
But still I have uploaded both the codes.
Please help me find my mistake.