Homework 10.1

Ans => Insertion in AVL tree is carried as follows -

- Search for the position where element should be.
- Insert the node.
- This may upset the balancing property.
- We traverse up the tree till we reach root or the node where imbalance occurs ~~and do the rota~~
- Do the rotation if imbalance is there and return.

Note : Once fixed, we don't need to go up till root as original height of balanced tree is always retained.

- Actually $O(h)$ but h is itself $O(\log n)$ for AVL tree.

$O(\log n)$ for searching and traversing up (if necessary)
$O(1)$ for inserting and reordering (if necessary)

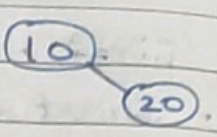
=> Total complexity $O(\log n + 1 + \log n) = O(\log n)$

Page No.
Date

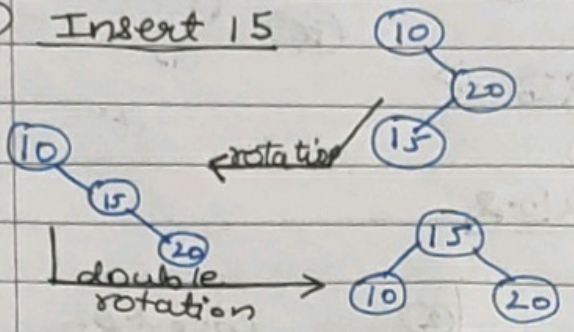
Homework 10.2

Ans=> ① Insert 10

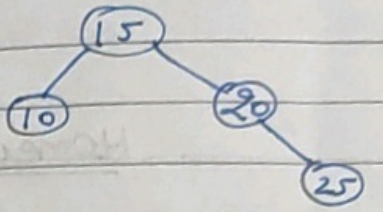
② Insert 20



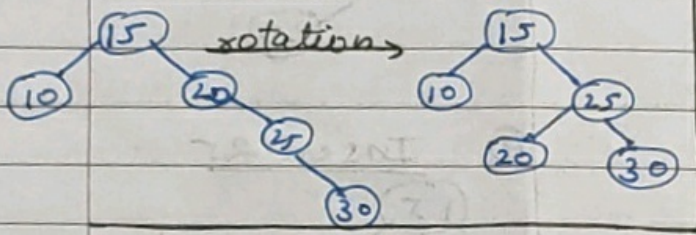
③ Insert 15



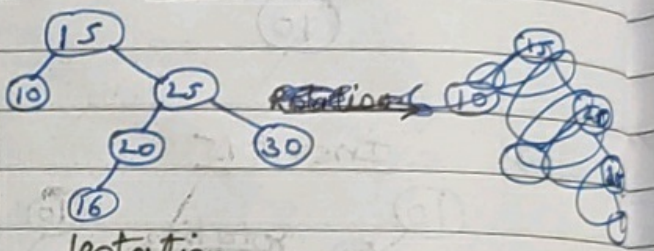
④ Insert 25



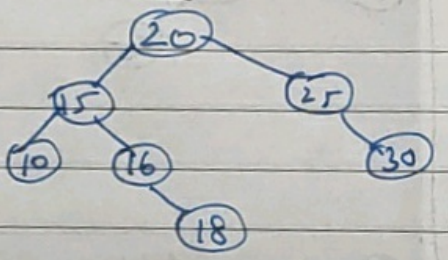
⑤ Insert 30

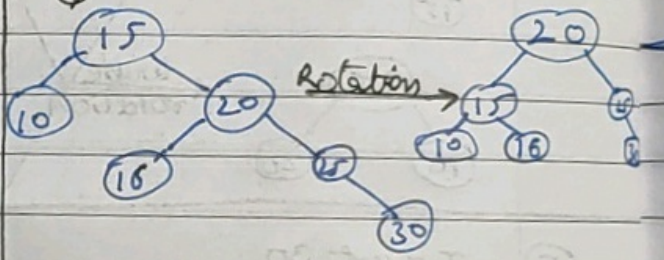


⑥ Insert 16

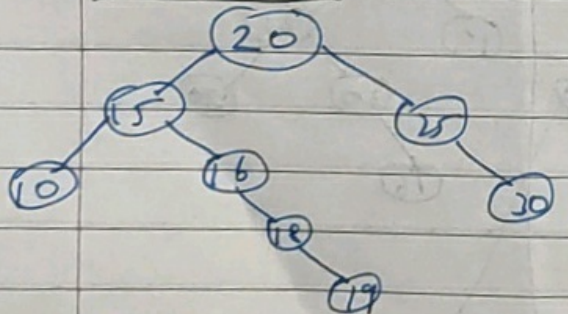


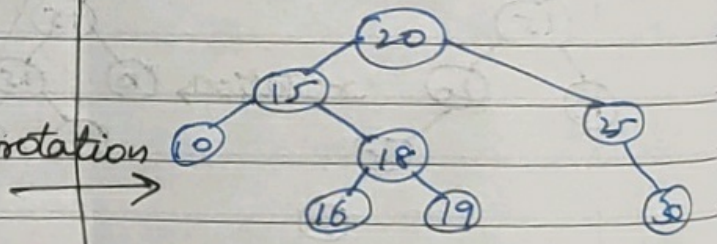
⑦ Insert 18 :





⑧ Insert 19 .

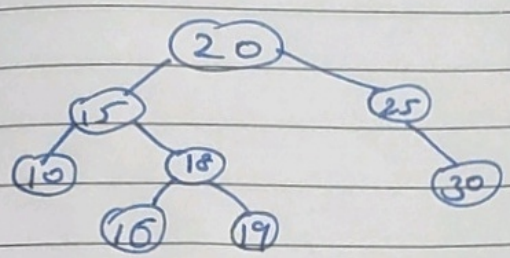




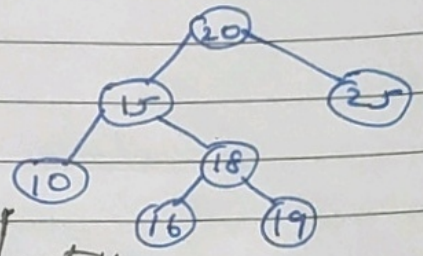
Page No.
Date

Homework 10.3

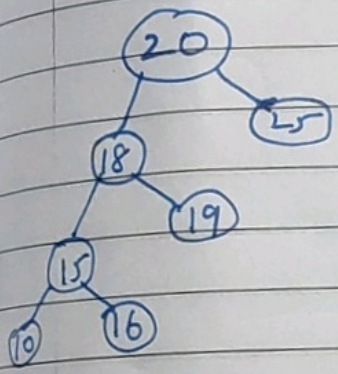
Ans) Tree =>



After deleting



rotation



rotation →

Final balanced AVL tree