4/4/2020 Data Structures

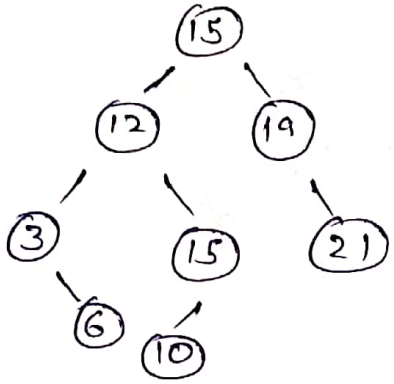
P. Rutwik 19BCS084 Sec:-B

① Inorder :- A, K, B, J, C, L, D, E, H, G, F, I

Preorder :- L, K, A, J, B, C, I, H, E, D, F, G

Post order :- A, B, C, J, K, I, D, E, F, G, H, L

② (i)

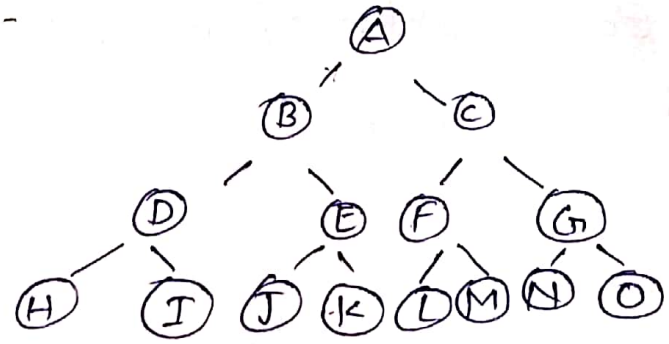


(ii) The tree is AVL tree [i.e, height is balanced]

③ Max nodes = 15

Min nodes = 4

Ex:-

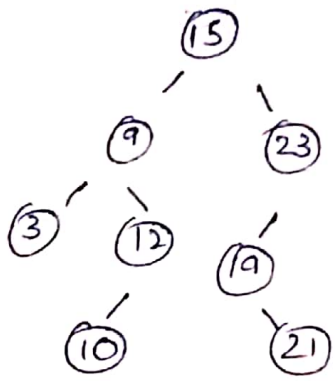


Internal nodes :- A, B, C, D, E, F, G

Leaf nodes :- H, I, J, K, L, M, N, O

④ False

The 1st item is not the smallest one. The binary tree that I considered is:



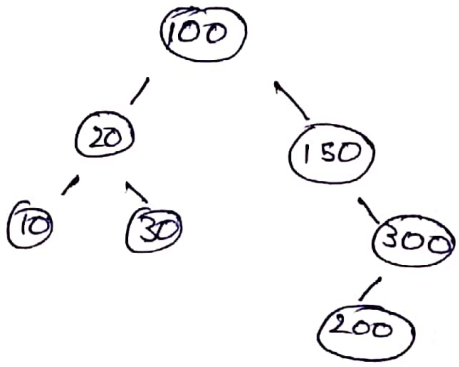
The above tree is a binary search tree in which the left node values of root are less than the root value and the right node value of the root is greater than the root value, the PREORDER for the above tree is 15,9,3,12,10,23,19,21. Since we can say that the 1st digit is not the least num -ber of the tree.

⑤ Level Order Traversal:-

2, 3, 5, 10, 8, 7, 22, 11, 13, 20, 24, 16

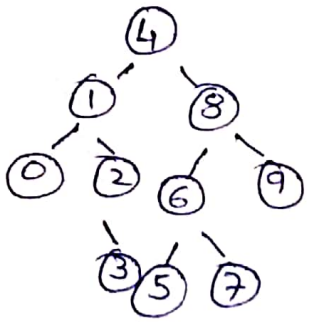
Initial array [] = {2,3,5,10,8,7,22,11,13,20,24,16}

⑥



⑦ 7,5,1,8,3,6,0,9,4,2

The binary search tree is



Inorder:- 0,1,2,3,4,5,6,7,8,9.