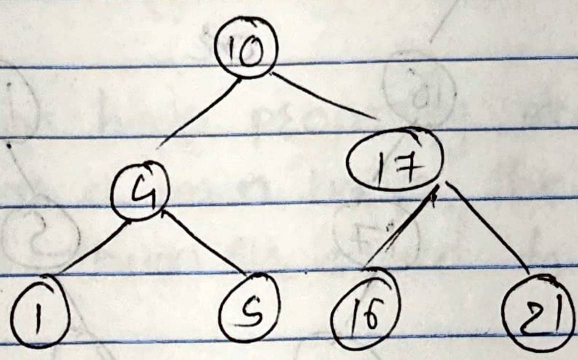
HW-10

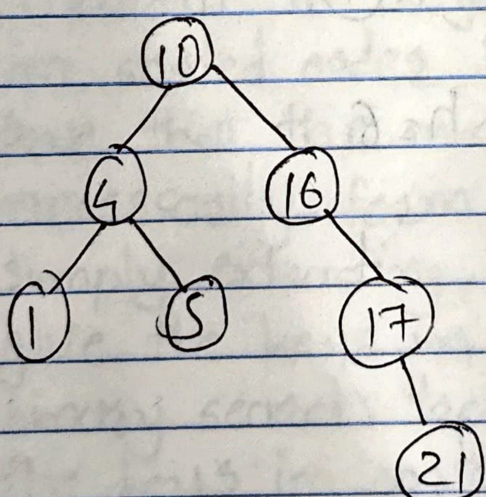
Neel Katrodiya 1002254987 DAA

ch-12

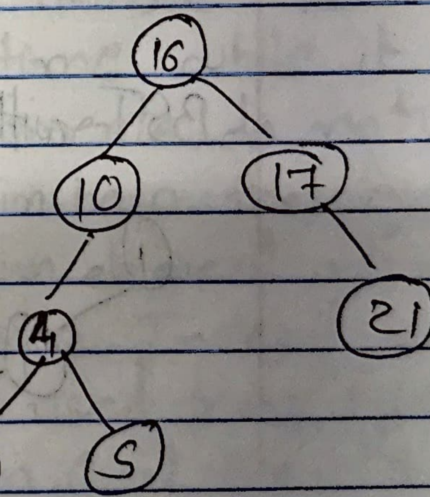
12.1-1 Binary search ~~the~~ tree with height $h=2$



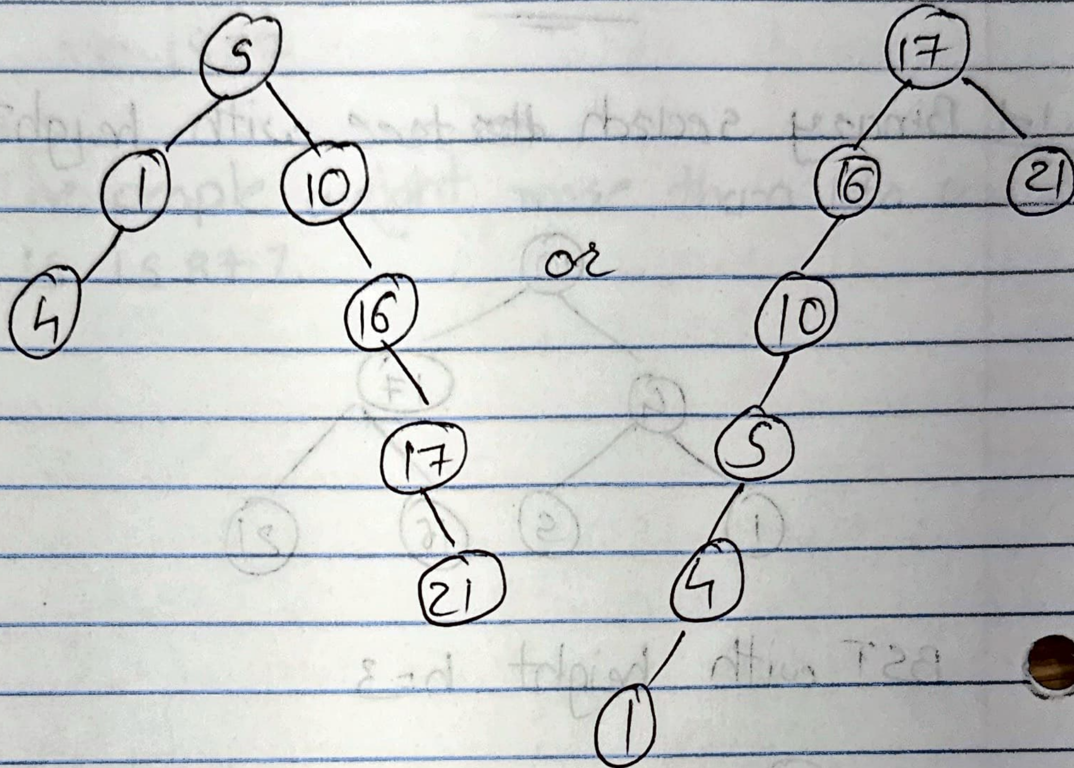
BST with height $h=3$



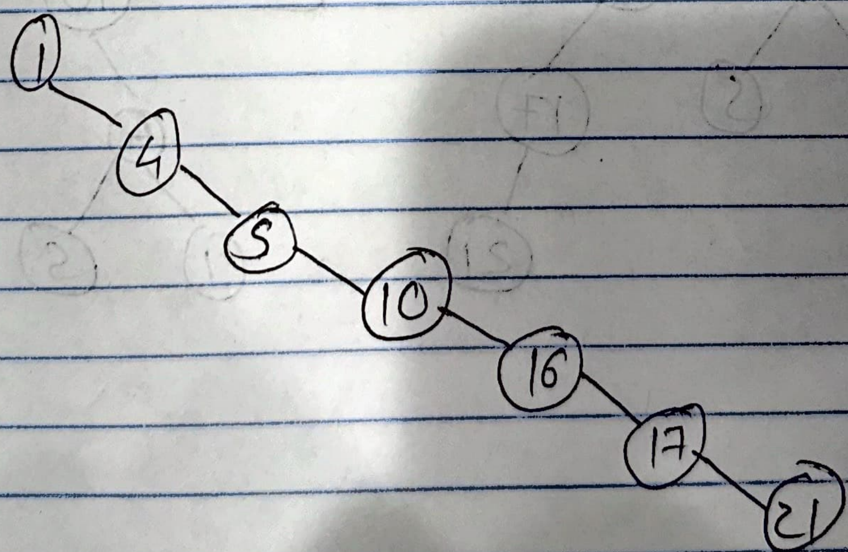
or



BST with height h=4



BST with height h=6

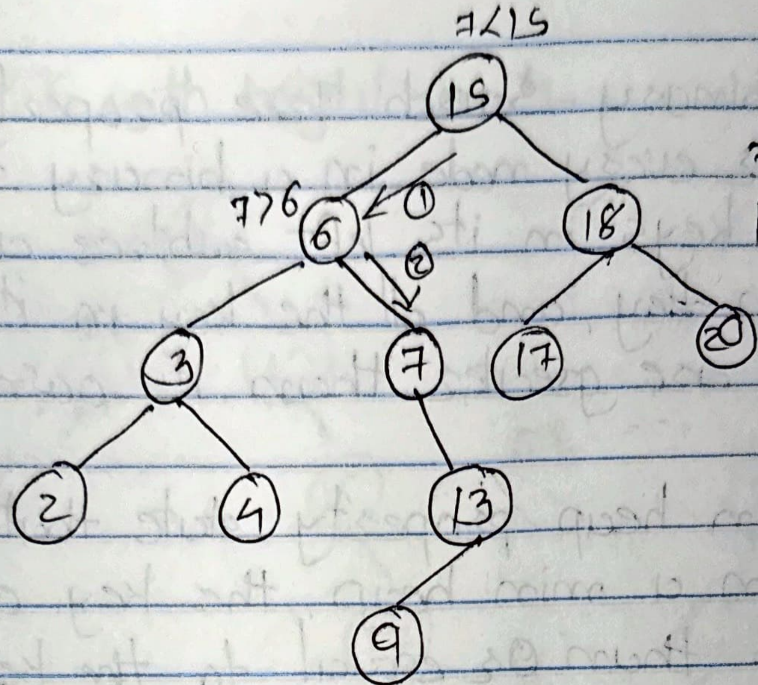


12.1-2 The binary-Search-Tree property state that for every node in a binary search tree, all the keys in its left subtree are less than its own key, and all the key in its right subtree are greater than its own key.

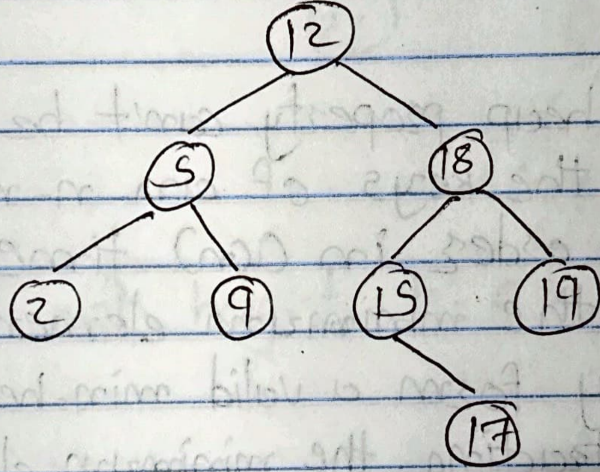
The min heap property state that for every node in a min heap, the key of the node is less than or equal to the key of its children.

The min-heap property can't be used to printout the keys of an n-node tree in sorted order in O(n) time. while it is true that the minimum elements do not necessarily form a valid min-heap. therefore, simply extracting, the minimum element will not give the keys in sorted order. Instead, a binary search tree can be used to printout the keys in sorted order in O(n) time by performing on in-order traversal of the tree

Q-1



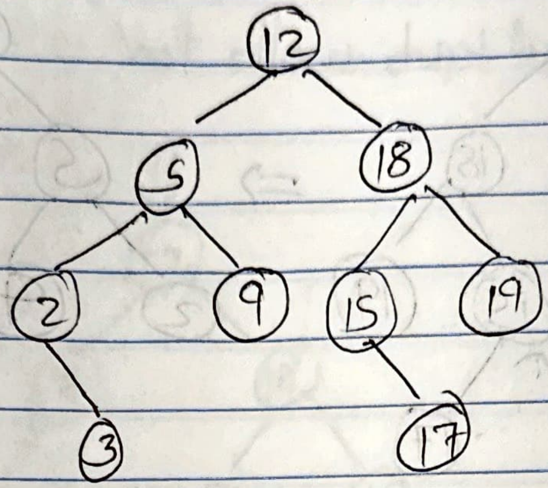
7<15 → move left
7>6 → move right
key 7 found



Q-2

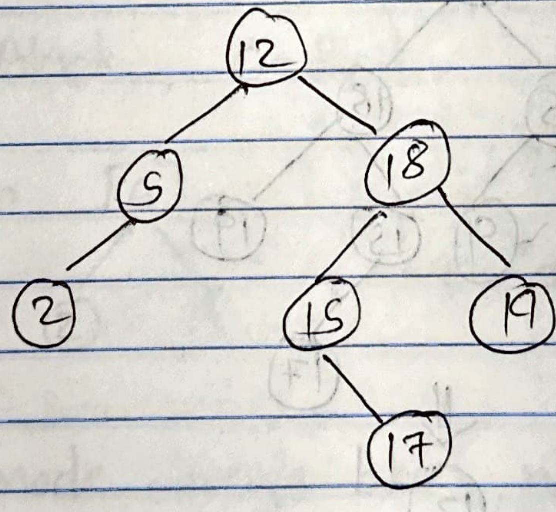
steps

3<12, move left
3<5, move left
3>2, move right
insert 3



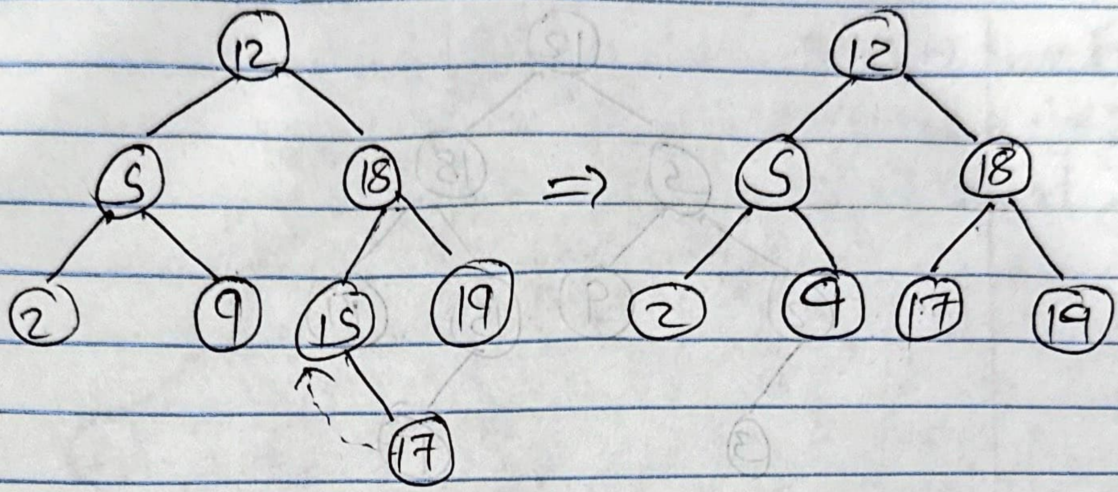
(3) a) delete key k=9

9 is leaf, it does not have any childnode so, it can be removed directly.



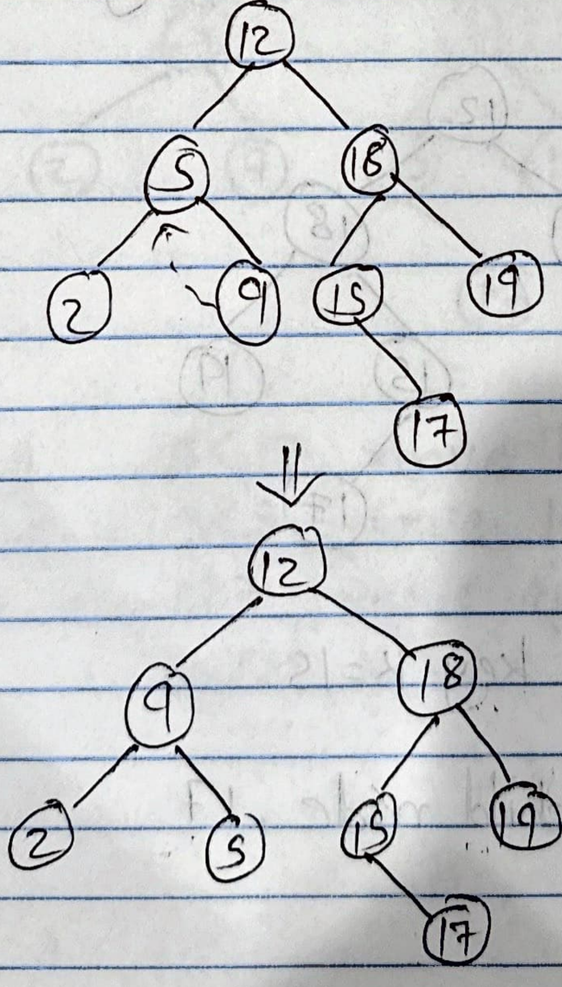
b) delete key k=15

15 has child node, 17

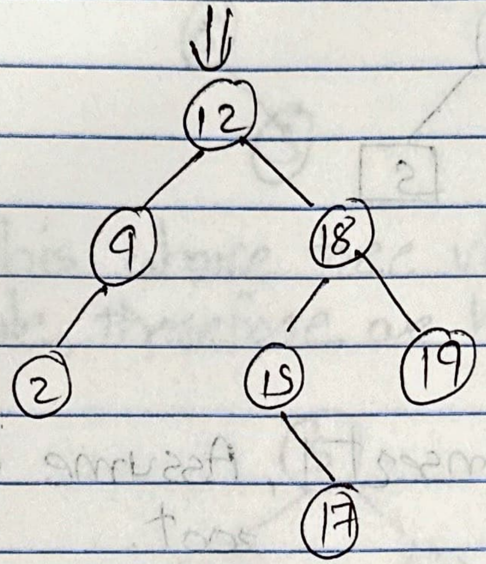


c) delete keys K=5

5 has 2 child node 2 & 9



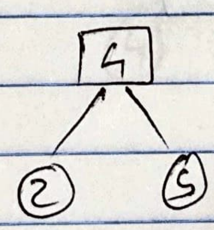
BST do not allow duplicates, so remove 9, then



Chapter - 13

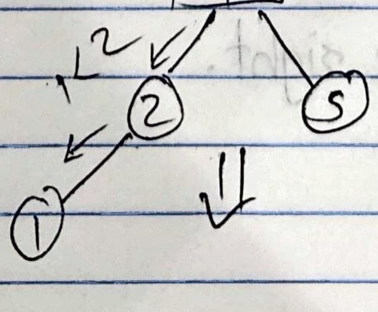
① □ = Black , O = Red

a) Given

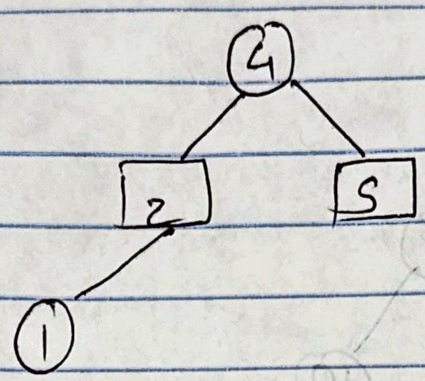


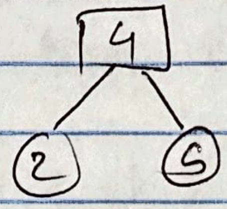
insert 1, Assume 4 is not root.

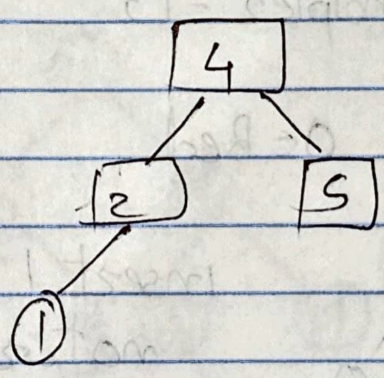
→ Root node always black, new element inserted as red.

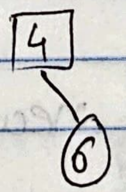


but two red nodes are not possible

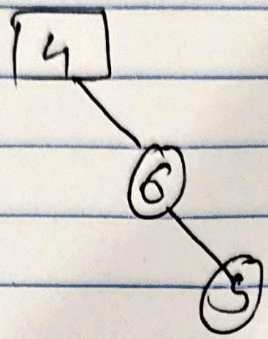


B) Given  insert 1, Assume 4 is root.



C) Given  insert 5, Assume 4 is not root.

→ 5 > 4, move right.



This above tree violates red-black tree rule, therefore we have to rotate it,