Vardaan
2024602

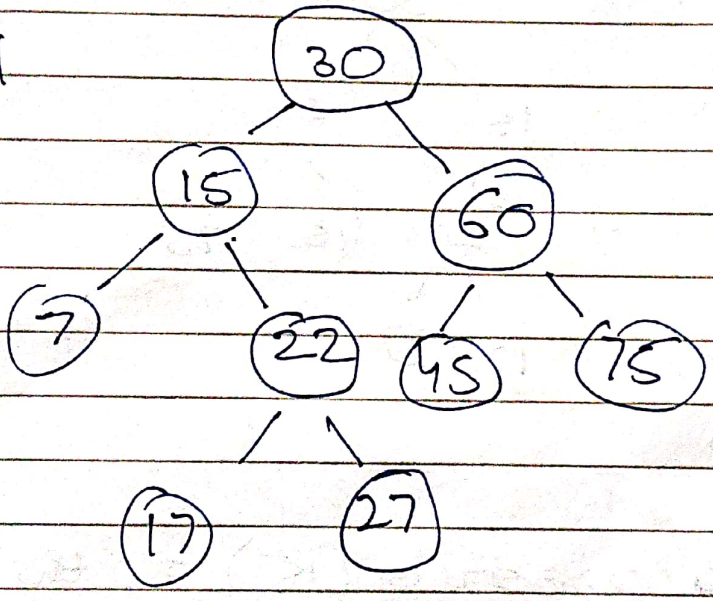
Date / /
Page No.

①

Assignment 3A

1) 30, 15, 60, 7, 22, 45, 75, 17, 27

1.1



1.2

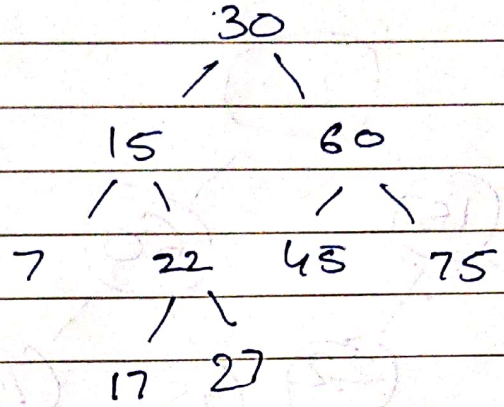
in order (L, Root, R)
→ 7, 15, 17, 22, 27, 30, 45, 60, 75

Pre order → (root, L, R) 30, 15, 7, 22, 17, 27, 60, 45, 75

Date / /
Page No.

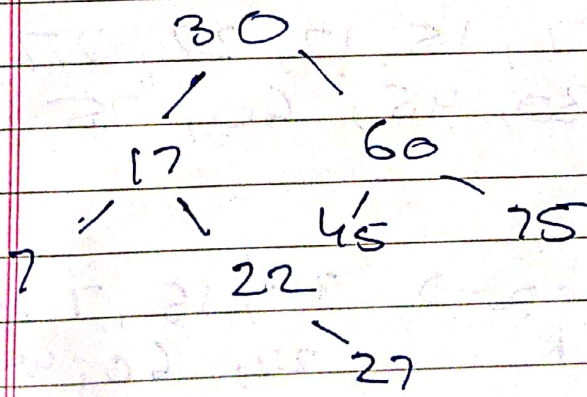
Post order (L, R, Root) → 7, 17, 27, 22, 15, 45, 75, 60, 30

1.3.



Successor of (15) = the left most in the right subtree of 15

⇒ 17



now we delete 15 and replace it by 17 and the attach the left & right subtrees of 15 to 17.

Date / /
Page No.

(2)

2)

1 4

Alice

↓

2 ← 4

Bob Alice

↓

2 ← 4 ← 5

Bob Alice Carol

↓

1 ← 2 ← 4 ← 5

Dave bob Alice carol

5 1 Dawie

6 2 ~~8~~ ← 4 ← 5

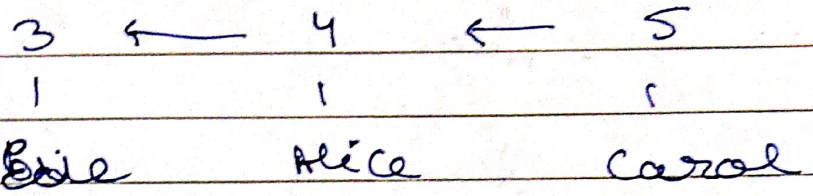
Eve

bob Alice Carol

7. 2 - bob

Date / /
Page No.

2.1



2.2.

In a binary tree the max height of the leaf from the root

$$= \lfloor \log(n) \rfloor + 1$$

where n is number of elements

for insertion we insert at the bottom and then heapify up.

for deletion we replace the node to be deleted by the leaf and then heapify down by this if we delete the root

Date / /
Page No.

then the max time it will take is $O(\log cn)$

Q2. 3.1 → code submitted seperately

3.2.

in usual BST

Right > root > left

but now

Right < root < left.

3.3.

in this approach
the time complexity
$O(n)$
& Space complexity
$O(\log n)$ $O(n)$

Space complexity is $O(h)$ where h is the height of the tree which is $\log n \leq h \leq n$

Date / /
Page No.

Q4.

4.2 → In this approch our time complexity = n log K

K → window ~~of the operation~~

n → number of elements

~~but if we use an~~ better approch

↓

$P_1 = 0$, $P_2 = Null$, $P_3 = Null$

~~i=0~~ i=1

for (i < n, i++) {

$P_1 = i$, $P_2 = i-1$, $P_3 = i-2$

if ($P_1, P_2, P_3 > 0$) {

return max (dis[P_1], Lis[P_2], Lis[P_3])

}

else continue

in this approch the time complexity = O(n)