

Binary Tree from Postorder :

Postorder: 9, 1, 2, 12, 7, 5, 3, 11, 4, 8 (LRV) → Root
Inorder: 9, 5, 1, 7, 12, 8, 4, 3, 11 (LVR)

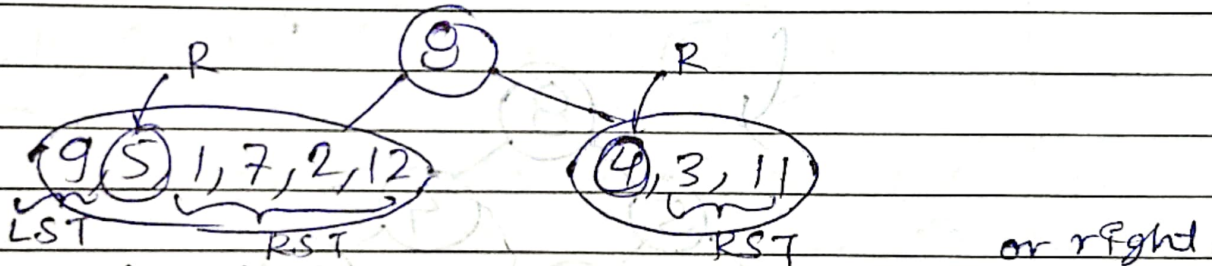
8

S-1: Find root in postorder i.e. the last element

S-2: Now locate the root in Inorder Trav.

S-3: Now all the elements to the left of root in In. trav. are left subtree & to the right are right subtree

S-4:



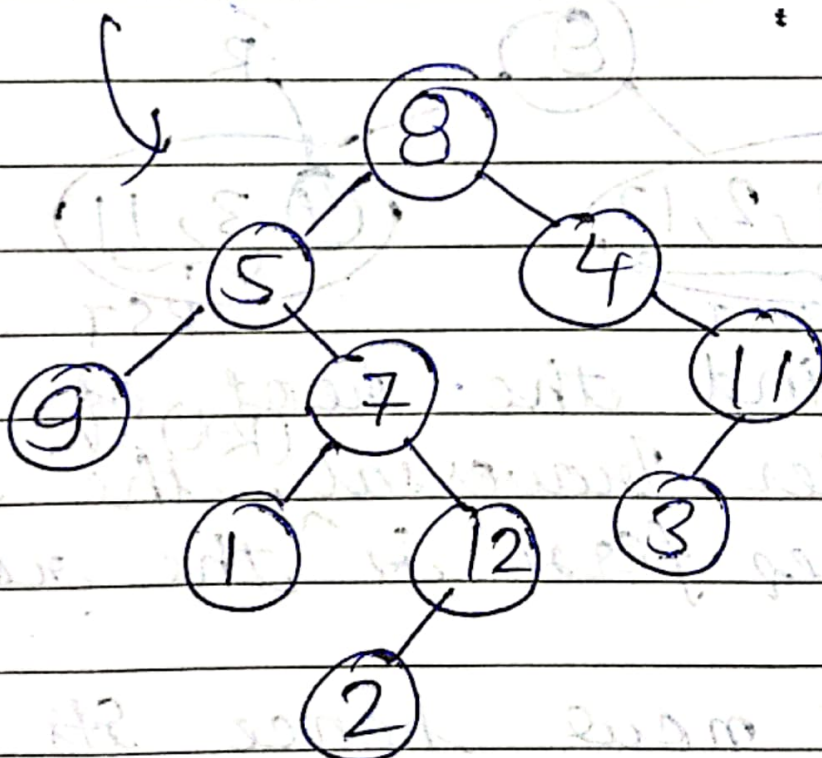
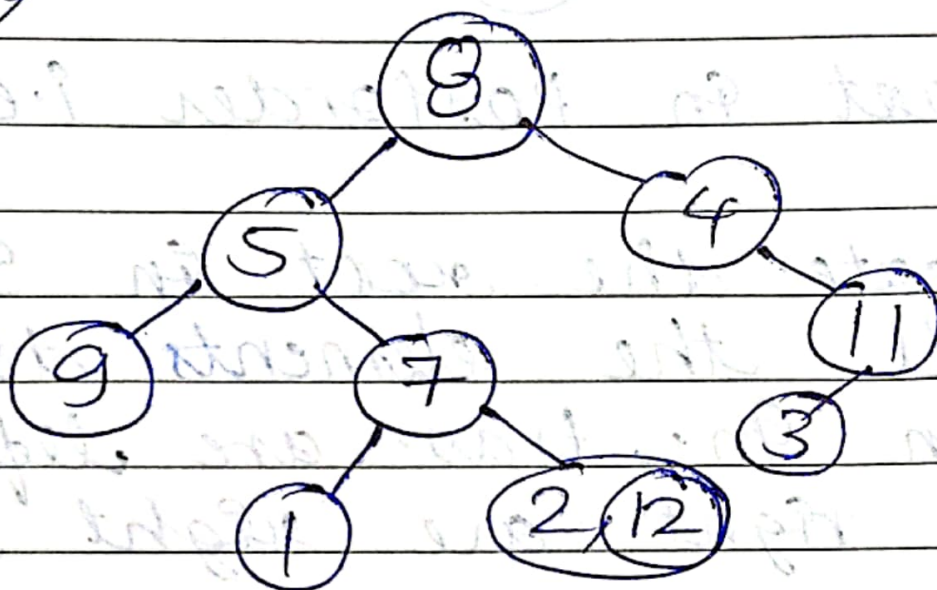
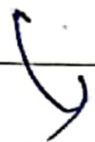
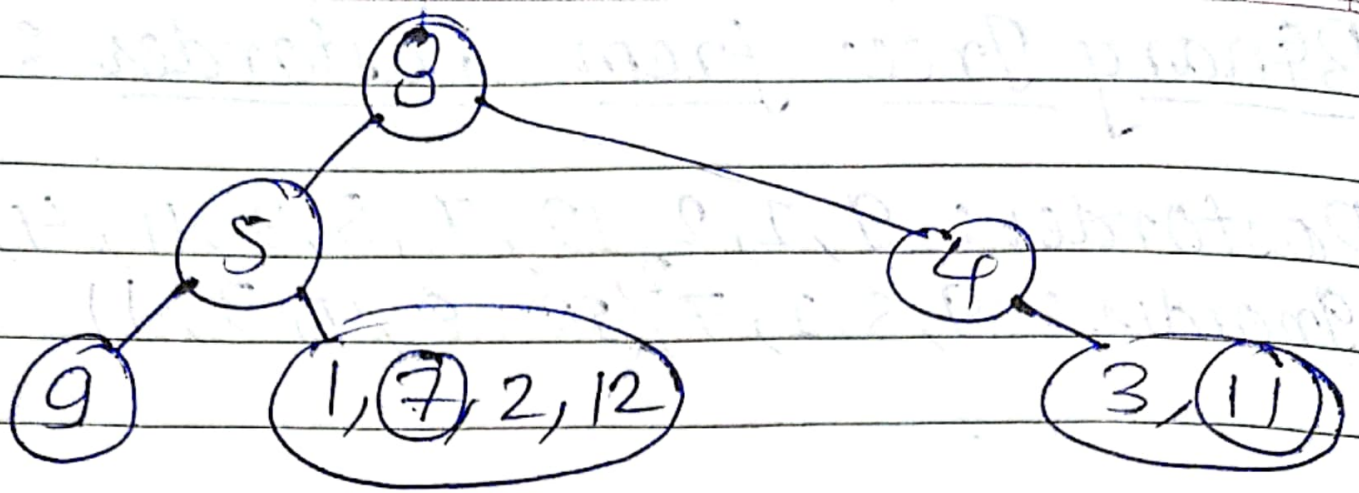
S-4: Now to find the root of left subtree, see Postorder Traversal, the element which comes first & the root

→ We got 5 now hence it's root of left subtree

→ Recur the process for entire tree

→ Now left subtree of 5 is 9 & RST is 1...12

→ Recur the process.



Binary Tree from Preorder

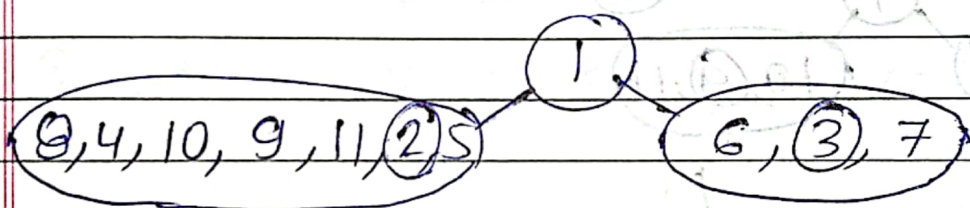
Preorder: 1, 2, 4, 8, 9, 10, 11, 5, 3, 6, 7 (VLR)

Inorder: 8, 4, 10, 9, 11, 2, 5, ① 6, 3, 7 (LVR)

S-1: Find root in Preorder i.e. the 1st element

S-2: Now locate root in the Inorder Traversal

S-3: In Inorder Traversal: all elements to the left of root are LST, & to the right of root are RST.

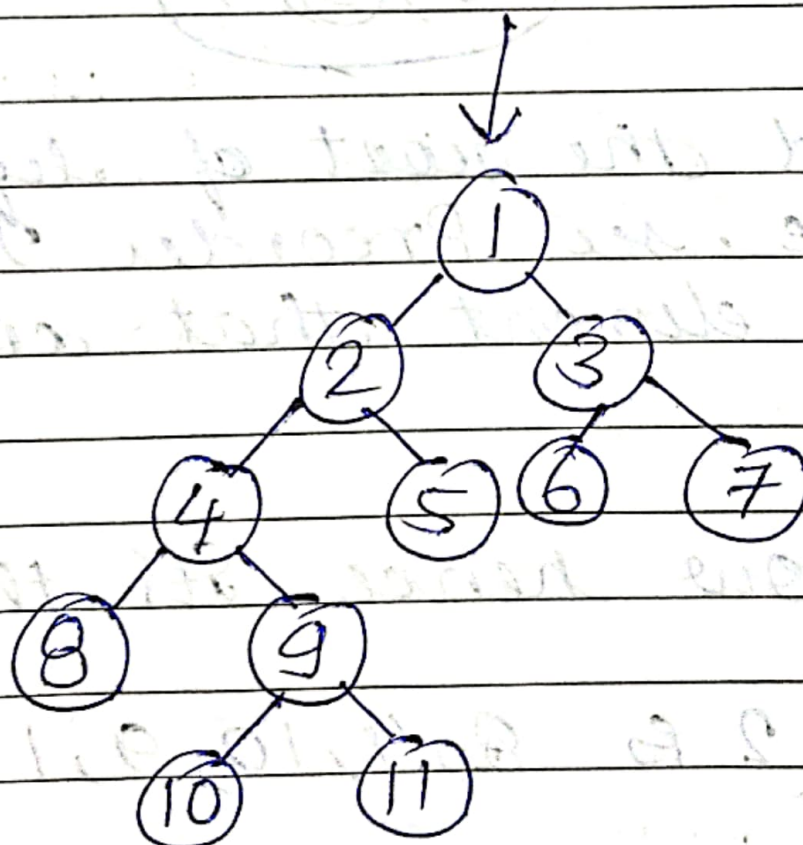
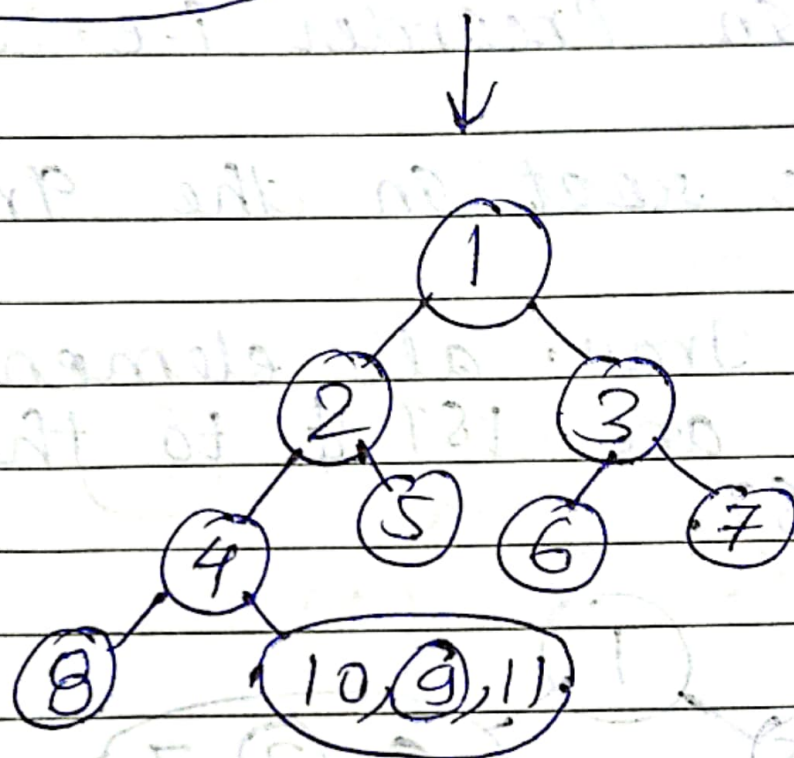
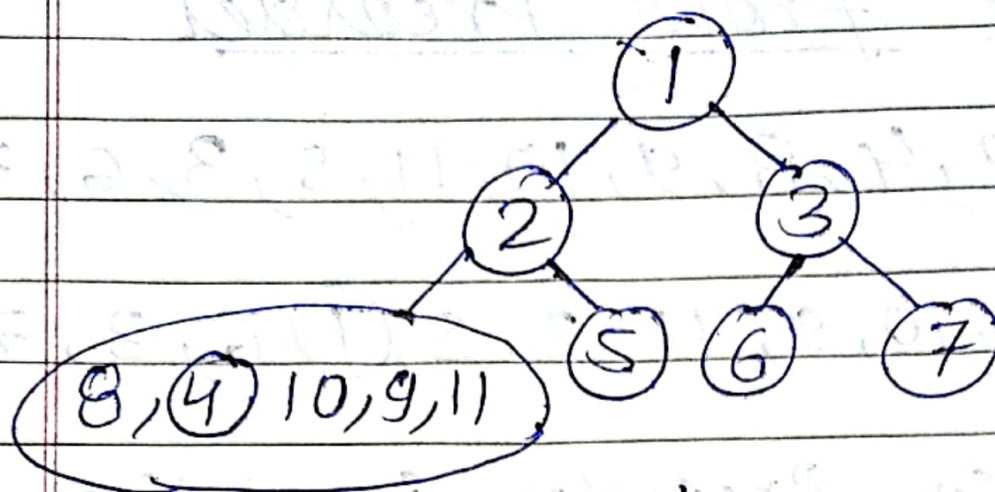


S-4: Now to find the root of left or right subtree, see Preorder Traversal (L → R), the element that comes first is the root

→ We got 2 now hence it's the root of LST

→ Now LST of 2 is 8, 4, 10, 9, 11 & RST of 2 is 5

→ Recur the process



Construct Binary Tree given Preorder & Postorder

Preorder: F B A D C E G I H

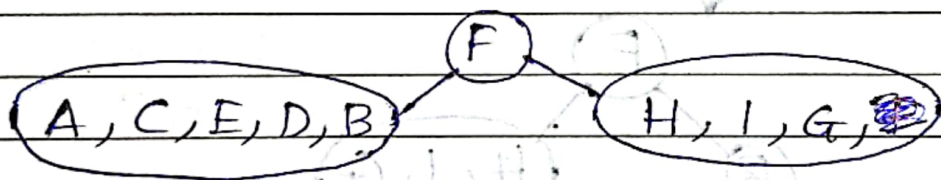
Postorder: A C E D B H I G F



S-1: Find Root from Preorder or Postorder (2nd elem)

S-2: Now see the next element after root in Preorder, here it's B

S-3: Look in Postorder, from starting to B is considered as LST of root & rem. elements are of RST

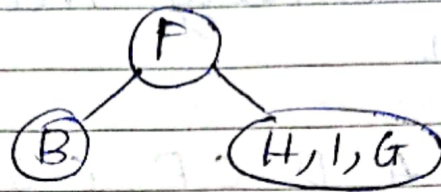


S-4: Now look at the second last element of Postorder, i.e. G

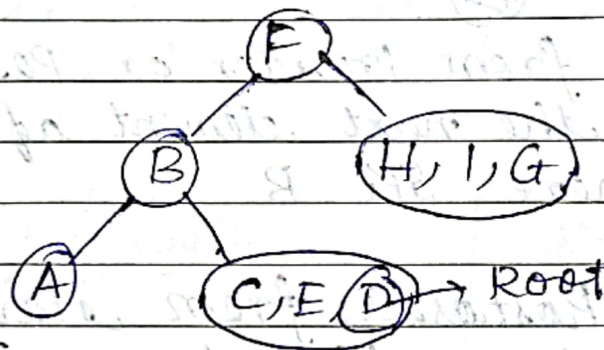
S-5: Look in Preorder, from B to G is the LST & from G to F is RST

S-6: Now Preorder of LST: B, A, D, C, E
Postorder " RST: A, C, E, D, B

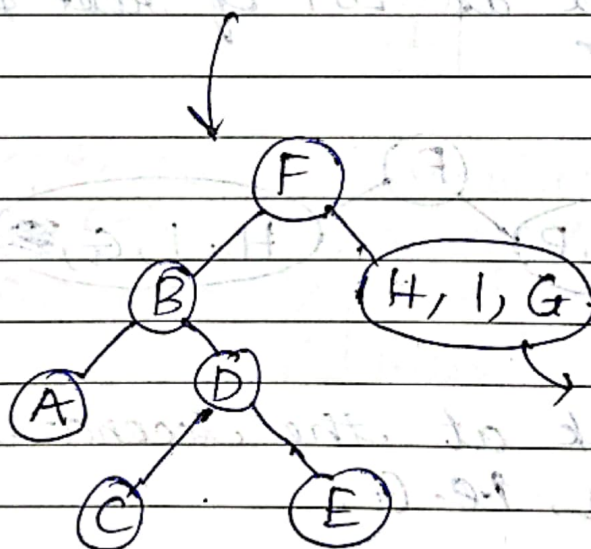
S-7: Now we can see root of LST is B



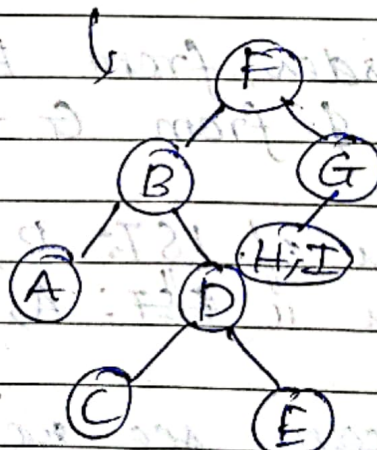
Now see 2nd element in Preorder & find LST & RST's from RST's Postorder

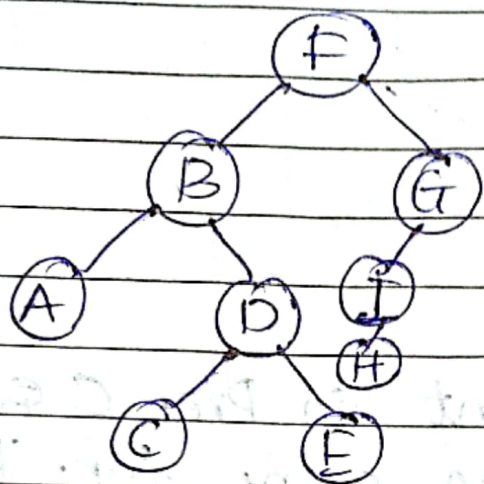


Preorder = B, G, E
Post = C, E, D

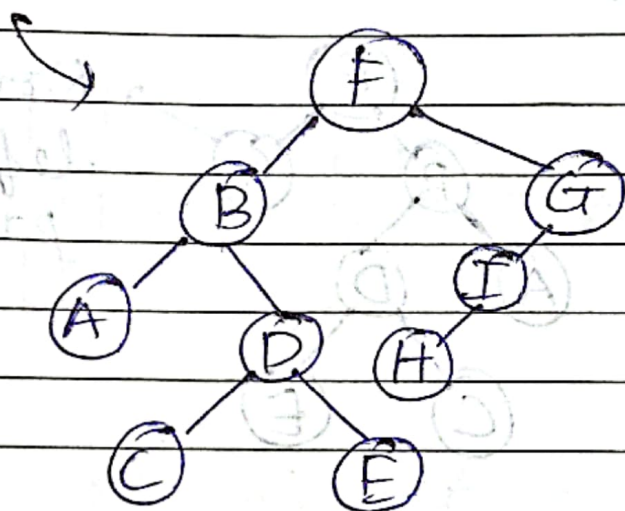


Poste → H, I, G
Pre = G, I, H





Pre = I H
Post = H I
LST

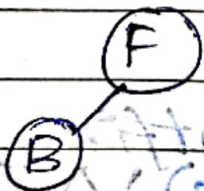


~~Trick:~~

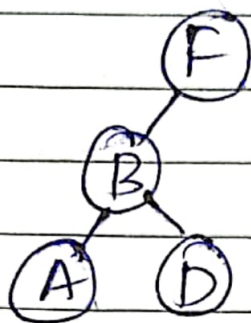
S-1: Find root



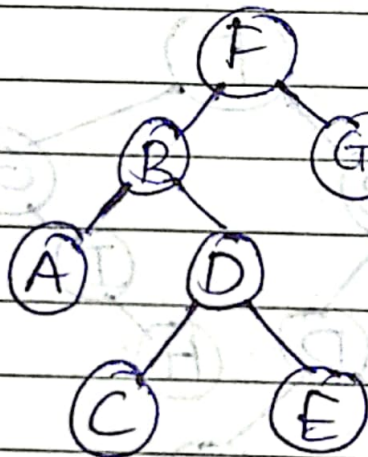
S-2: Now check: Preorder, 2nd element, i.e. B, as B is to the left of F in Post. It's the LST of F, hence part of F



S-3: Now check next ele. in Preorder, i.e. A, as A is to the left of B It's the 1st part of B & first full the LST of B



Now next element in Pre. C goes to the left of ~~B~~ D, also same for E



Now G is to the left of F hence part of F

Now I is to left of G \therefore part of G

