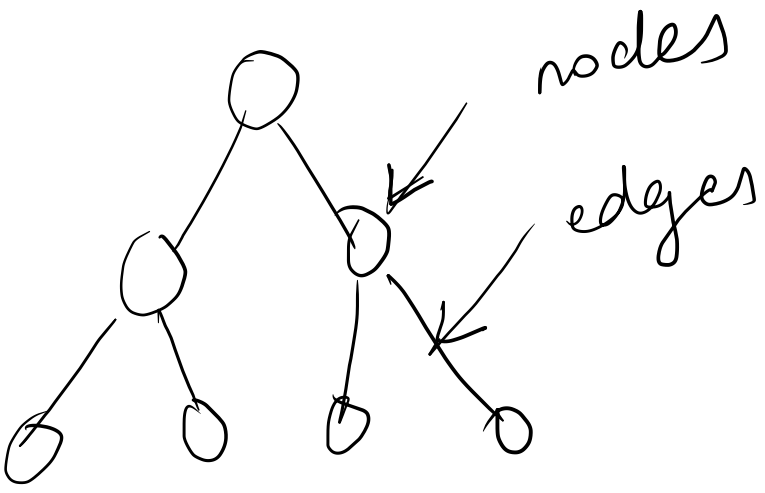
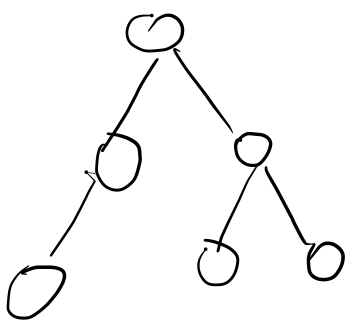
TREES → Non-linear

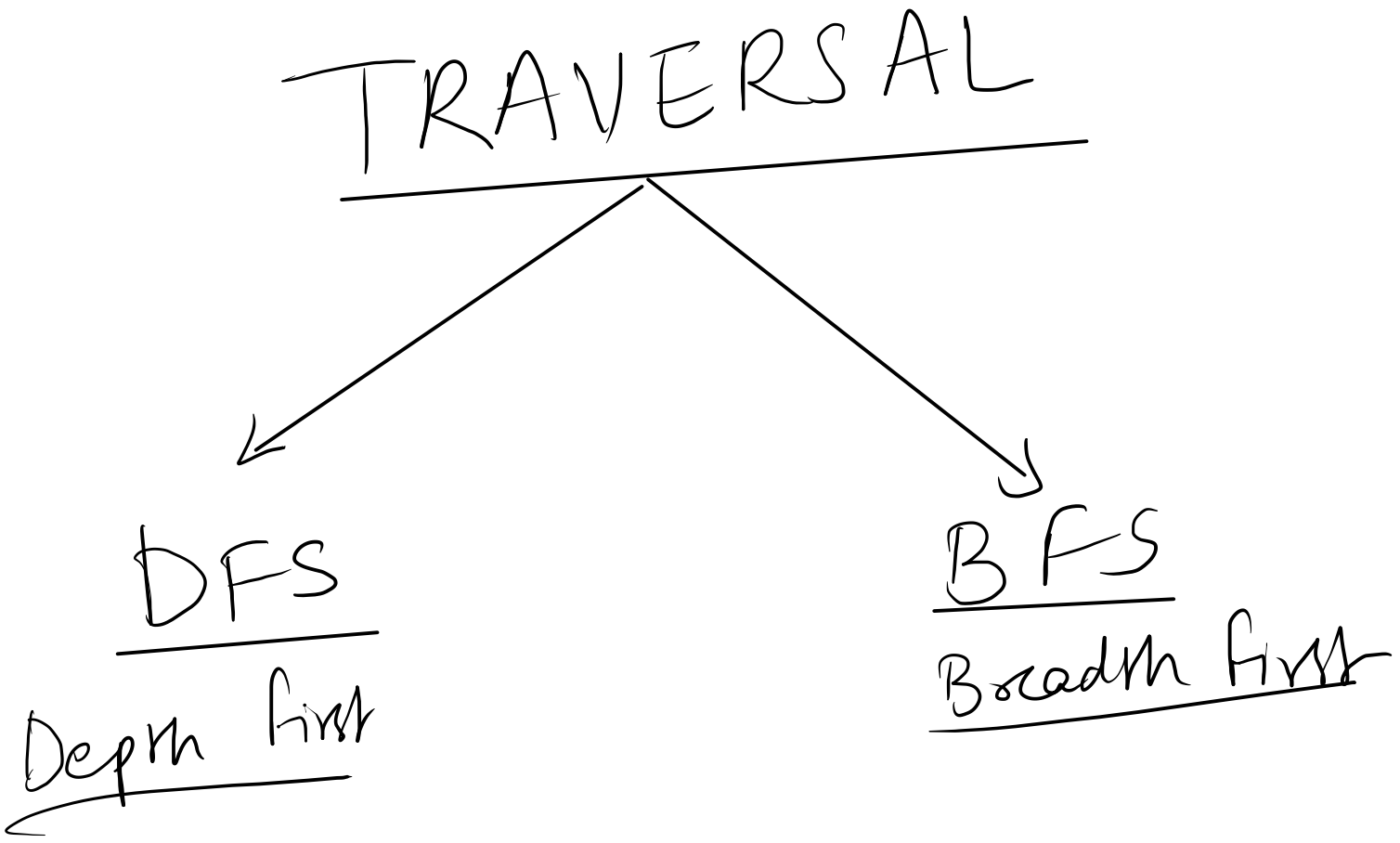


Binary Tree

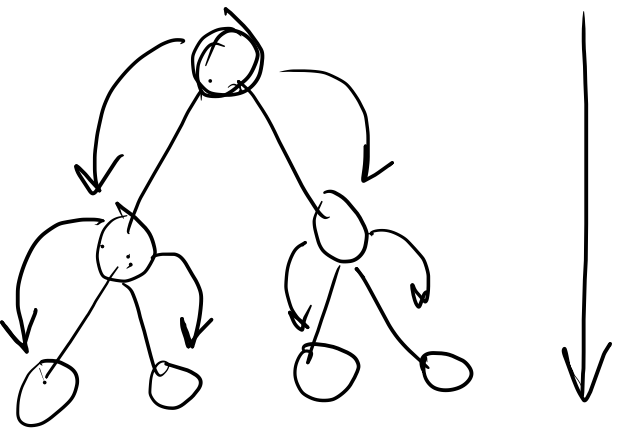
↳ max 2 child nodes

(0, 1, 2)



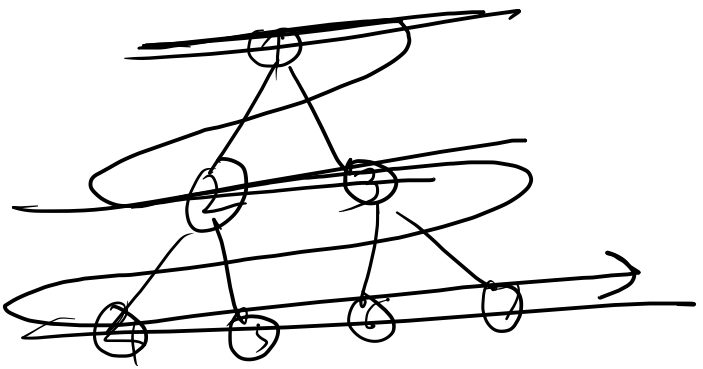

TRAVERSAL
DFS
Depth first
BFS
Breadth first

DFS



Depth

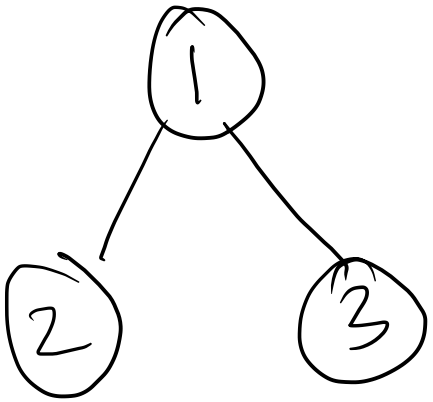
BFS

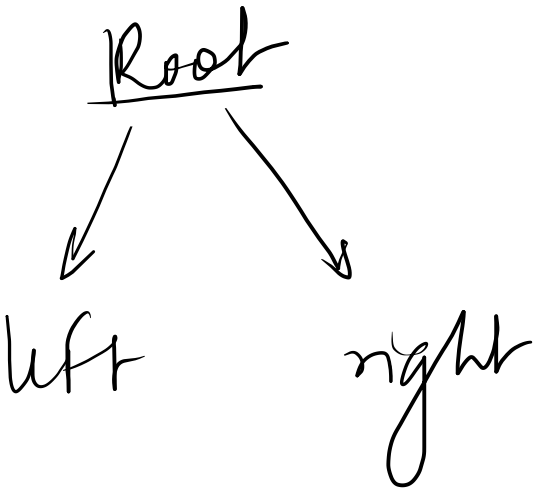


Breadth

DFS

1. Inorder
2. Preorder
3. Postorder


1
2
3


Root
left
right

Left Right

① Preorder

↳ root pehle ayega

→ Root, left, right

2

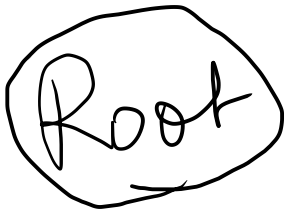
Inorder

↳ root bich ma ayega

Left Right

↓

Left Root Right



2 Postorder

root last ma ayega

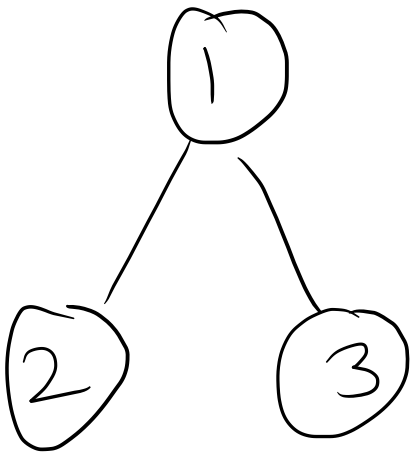
Left Right

Left Right Root

Preorder Root — —

Inorder — Root —

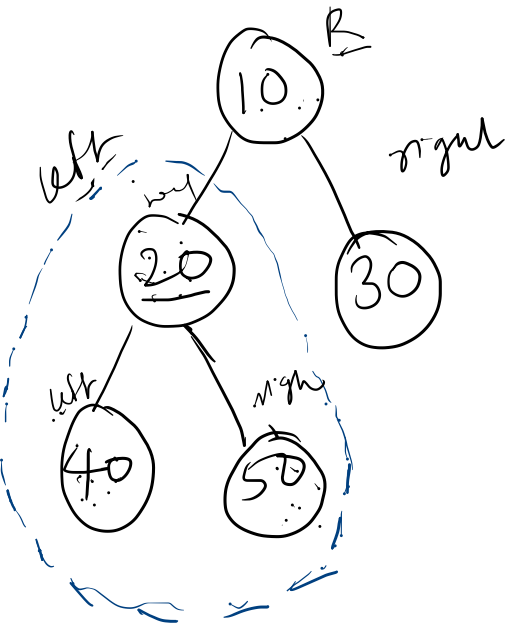
Postorder — — Root



(A) Preorder : 1, 2, 3

(B) Inorder : 2, 1, 3

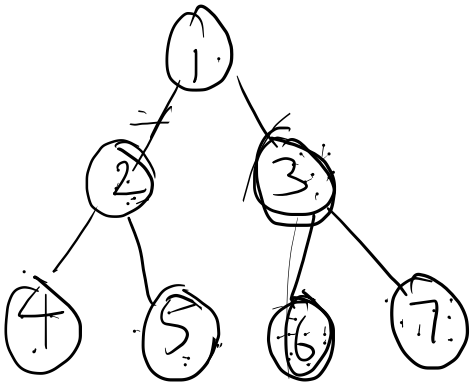
(C) Postorder : 2, 3, 1



(A) Preorder : Root, Left, Right
10, 20, 40, 50, 30

(B) Inorder : Left, root, right
40, 20, 50, 10, 30

(C) Postorder : left, right, root
40, 50, 20, 30, 10



(A) Preorder

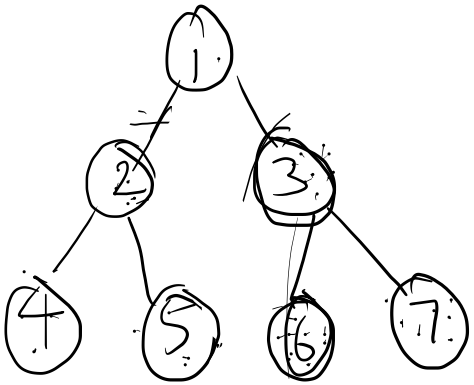
Root	Left	Right
1,	2, 4, 5,	3, 6, 7

(B) Inorder

Left	Root	Right
4, 2, 5,	1 ,	6, 3, 7

(C) Postorder

Left	Right	Root
4, 5, 2,	6, 7, 3,	1



(A) Preorder

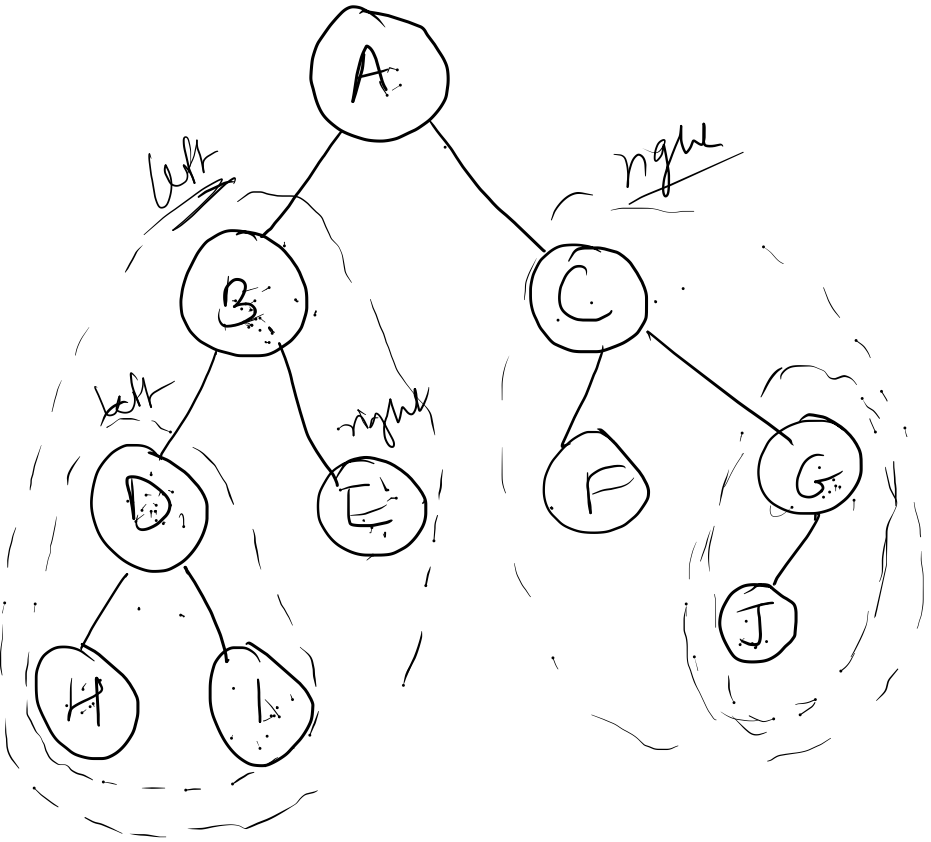
Root	Left	Right
1,	2, 4, 5,	3, 6, 7

(B) Inorder

Left	Root	Right
4, 2, 5,	1 ,	6, 3, 7

(C) Postorder

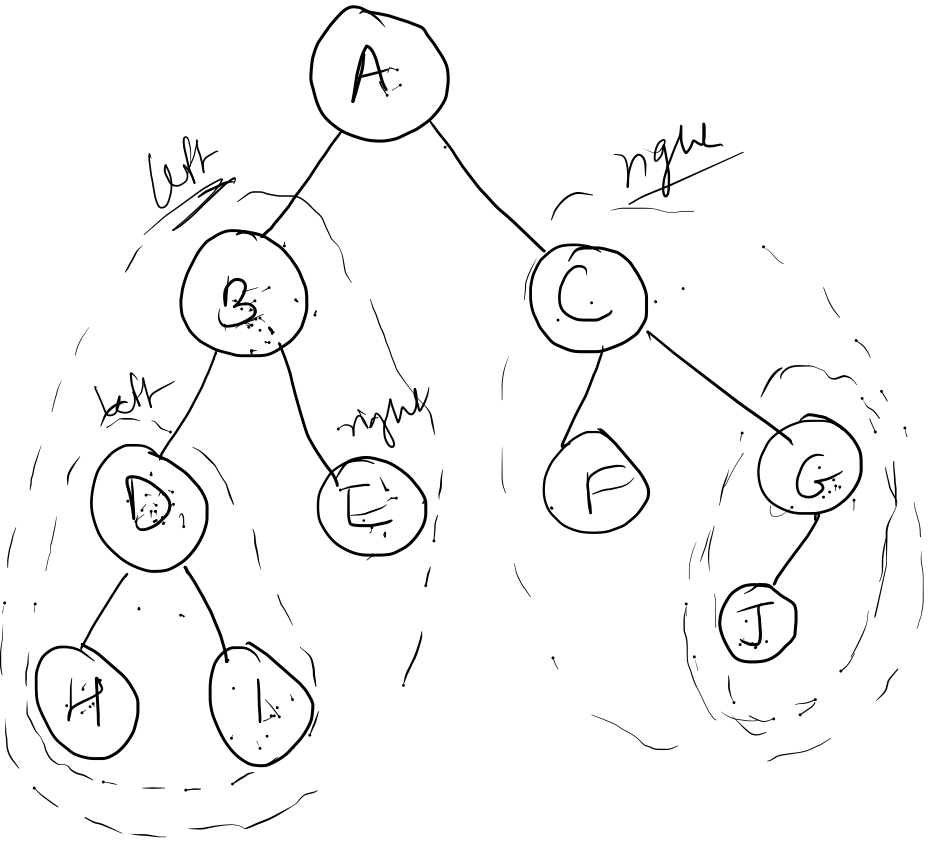
Left	Right	Root
4, 5, 2,	6, 7, 3,	1



PREORDER

Root, Left, Right

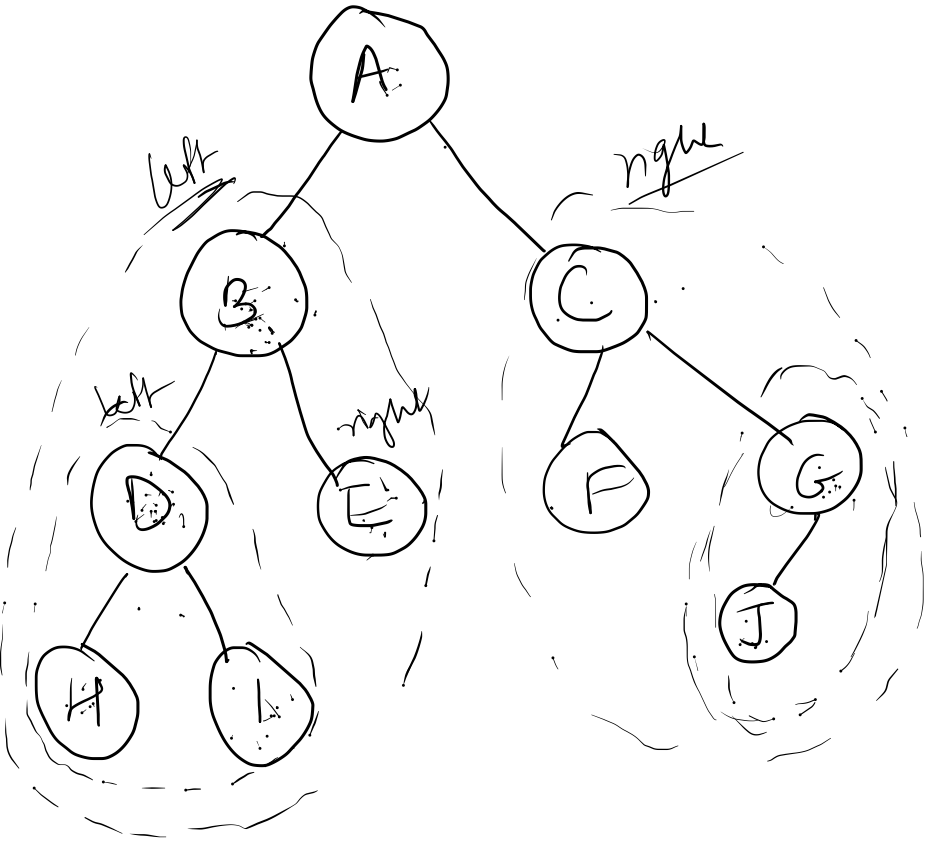
A, B, D, H, I, E, C, F, G, J



PREORDER

Root Left, Right

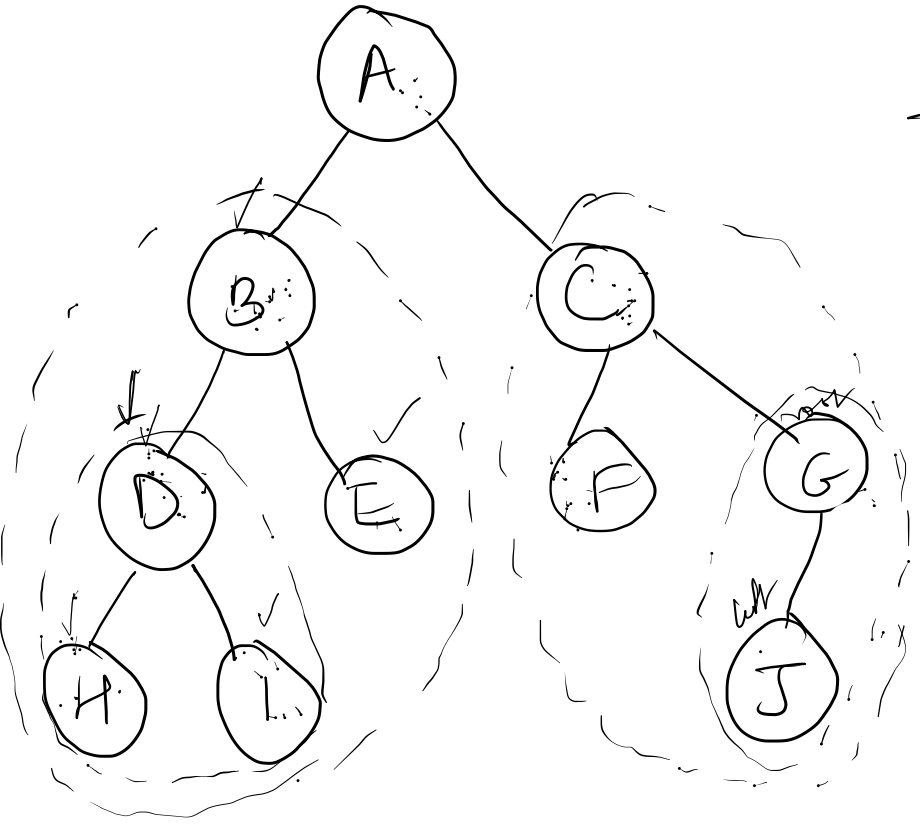
A, B, D, H, I, E, C, F, G, J



PREORDER

Root Left, Right

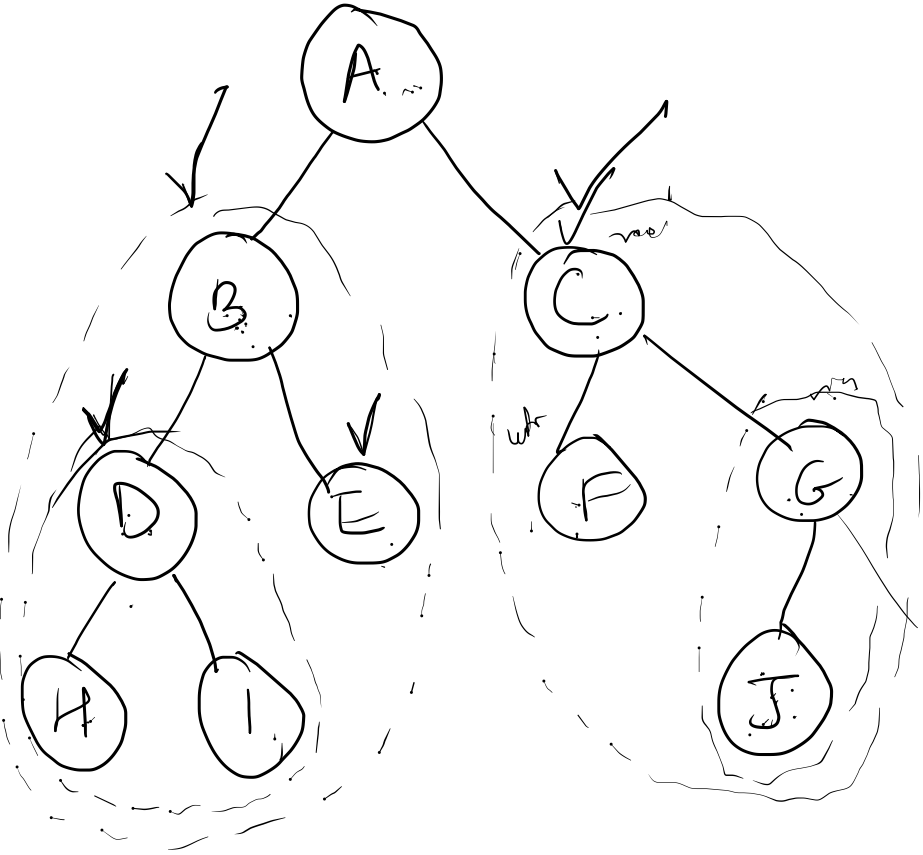
A, B, D, H, I, E, C, F, G, J



INORDER

Left, Root, Right

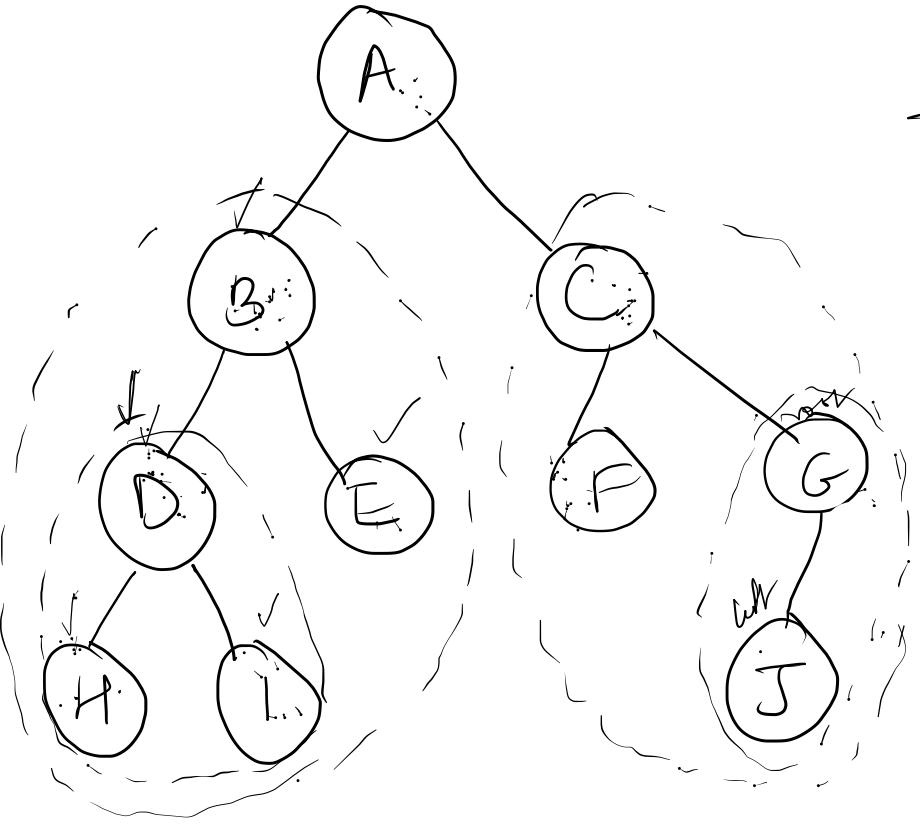
H, D, I, B, E, A, F, C, J, G



POST ORDER

Left , Right , Root .

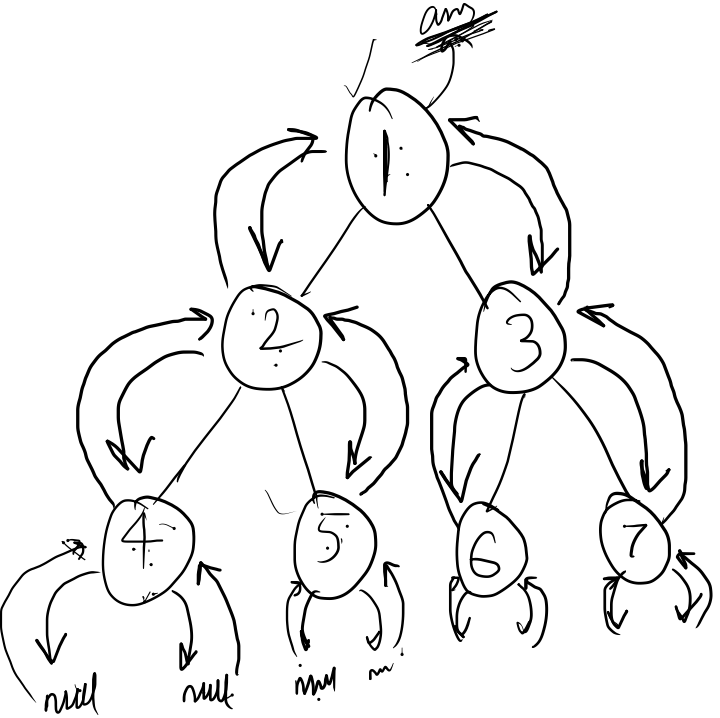
H, I, D, E, B, F, J, G, C, A



INORDER

Left, Root, Right

H, D, I, B, E, A, F, C, J, G



Preorder

Root left right

{ 1, 2, 4, 5, 3, 6, 7 }

if (node == null)
return

Add me
call left
call right