unacademy

Trees - Part IV

Course on Data Structure

Pankaj Sharma • Lesson 24 • Aug 4, 2024

unacademy

CS & IT Engineering

Data Structure

Tree

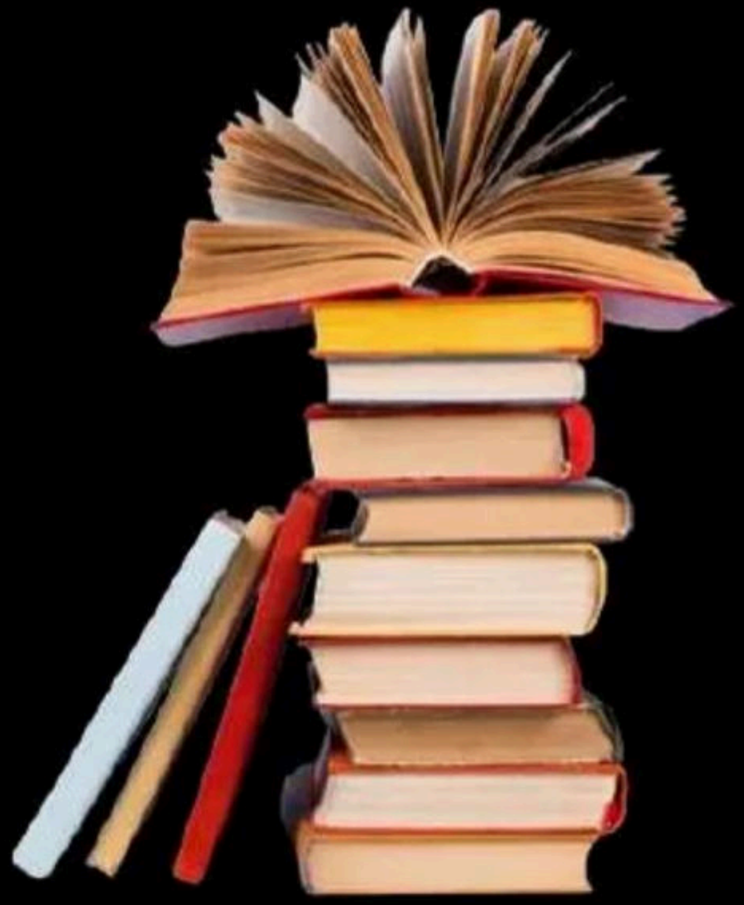
Lecture Number- 24

By- Pankaj Sir

unacademy

Topics

to be covered

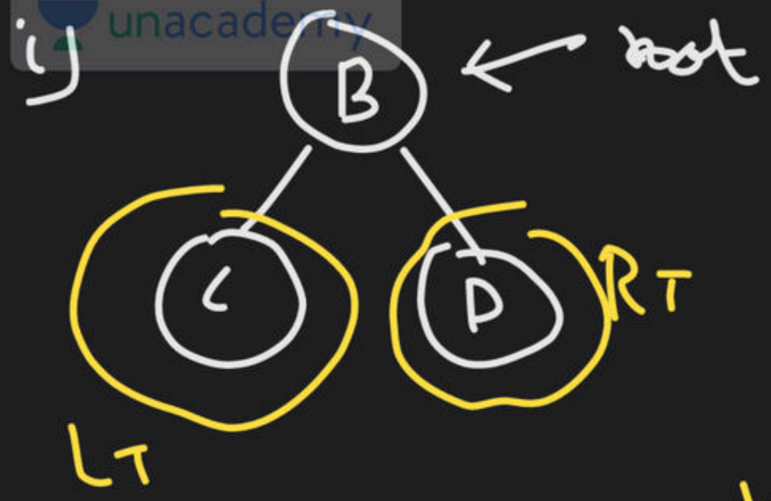


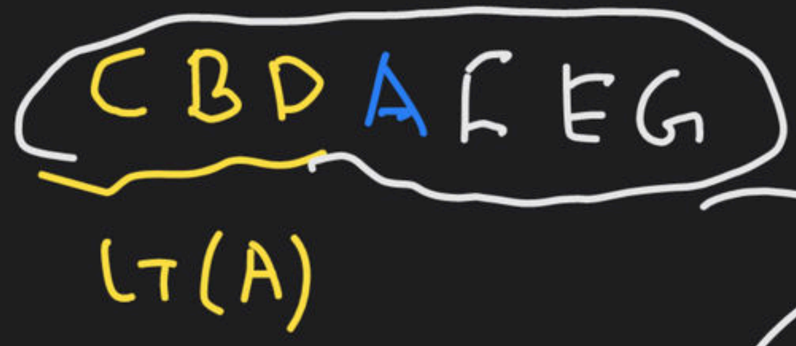
1. Tree-III

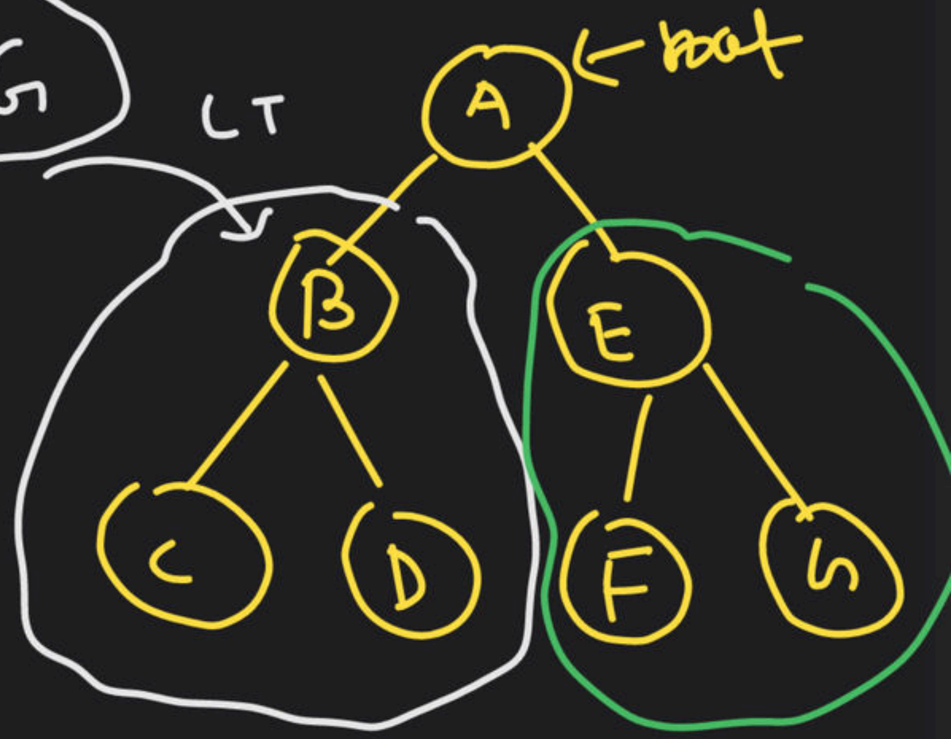
ABDEHCFG

Inorder Traversal

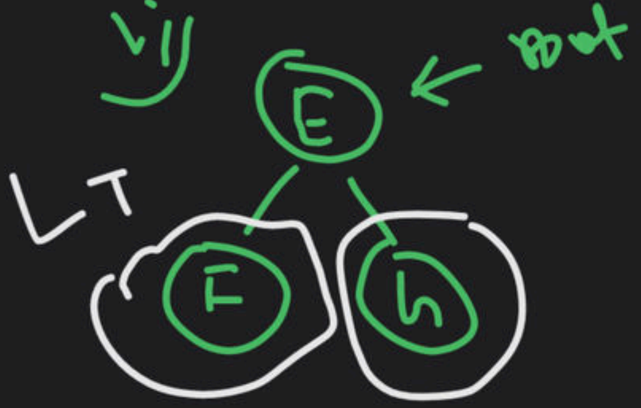
1) Traverse L_T of root node in Inorder
2) Print/Visit/Process root node
3) Traverse R_T of root node in Inorder.

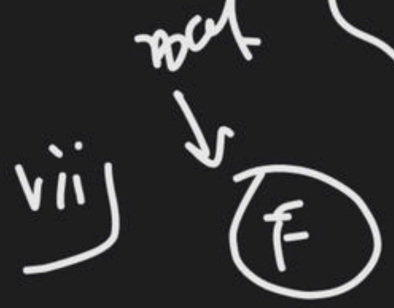






iv) Print A





iii) Print B

viii) Print E

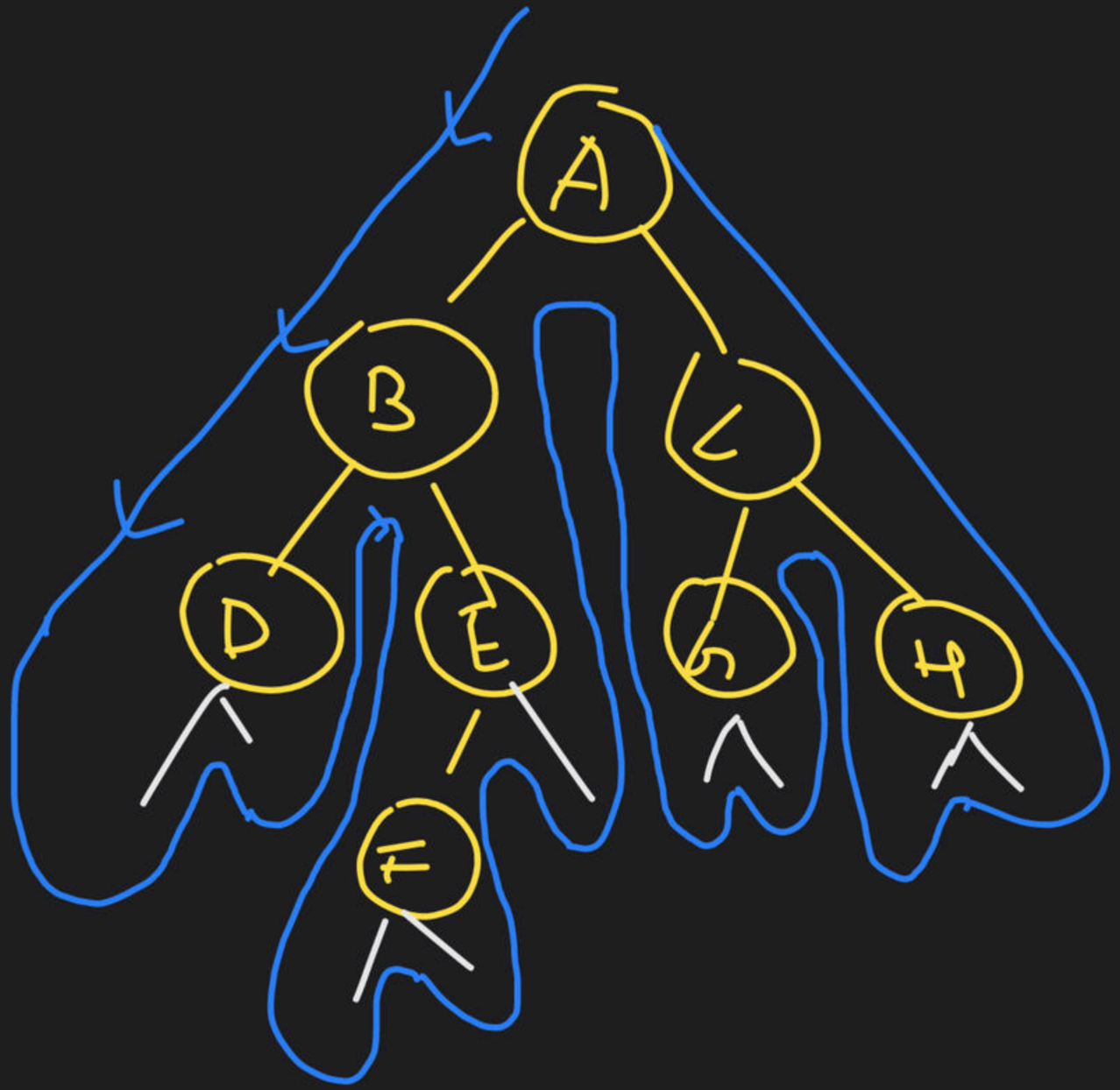
```
void Inorder(struct Node *Ptr)
{
    if(Ptr == NULL)
        return;
    Inorder(Ptr->Left);
    printf("%d", Ptr->data);
    Inorder(Ptr->Right);
}
```

unacademy

In: R

D B F E A G C H

L_T R_T

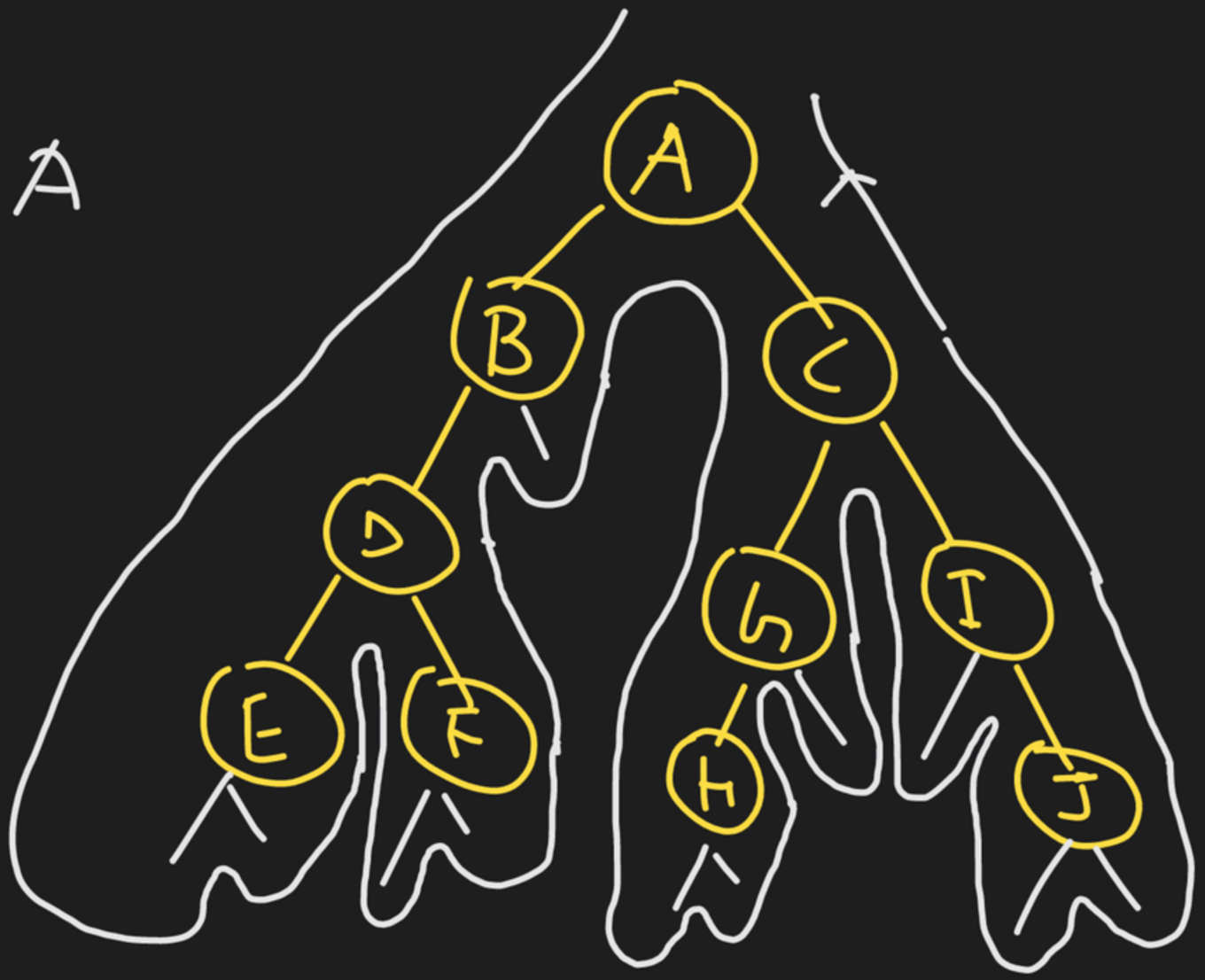


Post Order Traversal

1) Traverse L_T of root in Postorder
2) Traverse R_T of root in Postorder
3) visit/Print root node.

```
void Postorder (struct Node *ptr){
    if (ptr == NULL)
        return;
    Postorder(ptr->Left);
    Postorder(ptr->Right);
    printf("%d", ptr->data);
}
```

E F D B H G J I C A



unacademy

Unlabelled

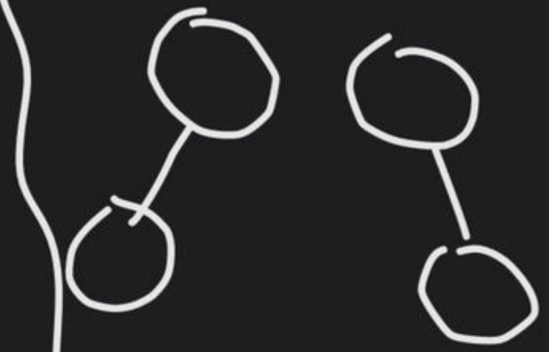
labelled

#unlabelled binary trees with n nodes
(shape / structure / topology)

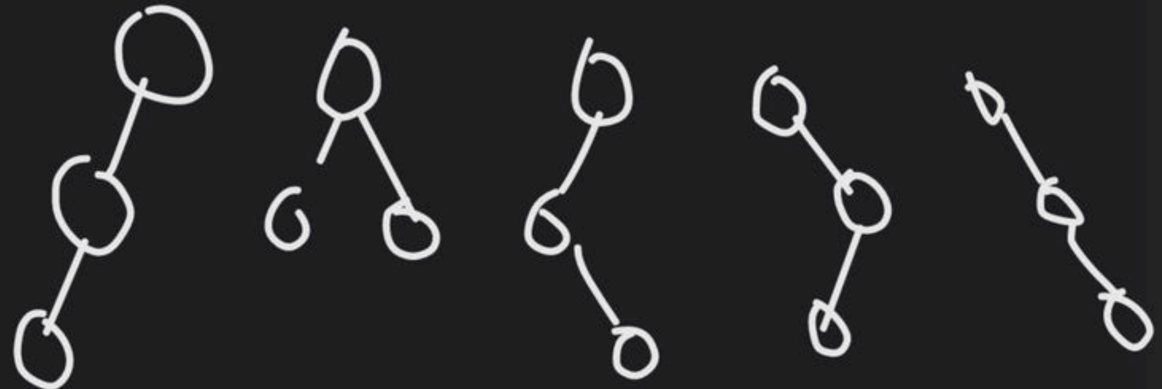
$n = 1$

1 ✓

$n = 2$



$n = 3$



5 ✓

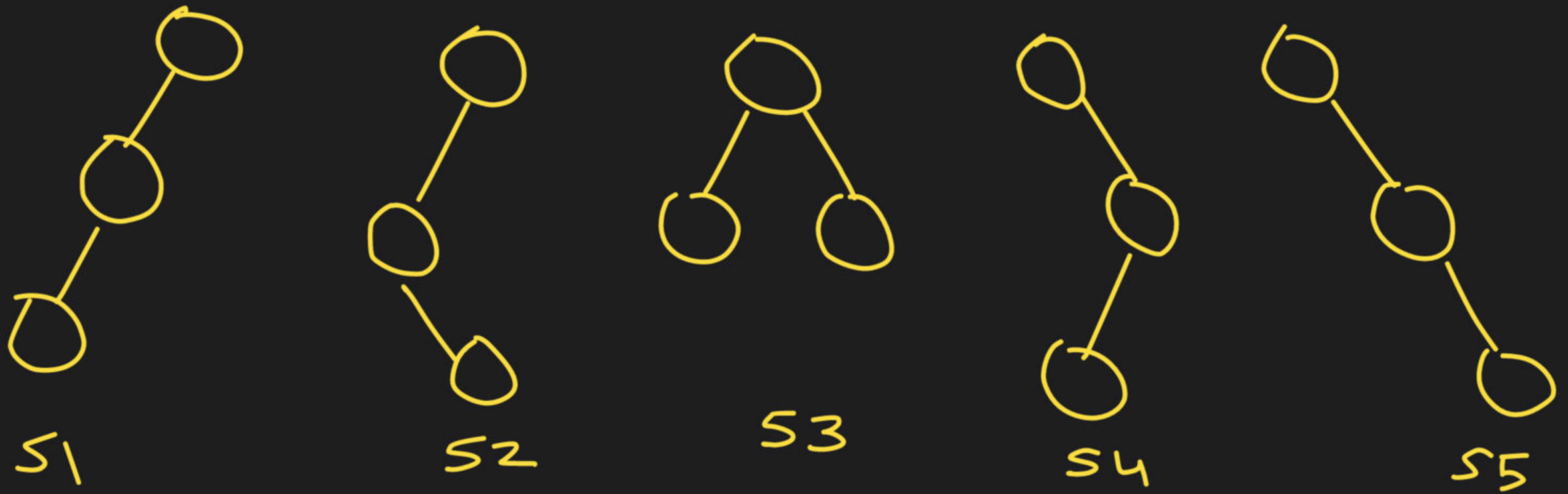
#Unlabelled binary tree with n nodes $= \frac{{}^{2n}C_n}{n+1}$

$n = 3$

$$\frac{{}^{6}C_3}{3+1} = \frac{{}^{6}C_3}{4} = \frac{1}{4} \times \frac{6!}{3!3!} = \frac{1}{4} \times \frac{6 \times 5 \times 4 \times 3!}{3! \times 3!}$$

$$= 5$$

$n = 3 \Rightarrow 5$ structure are possible

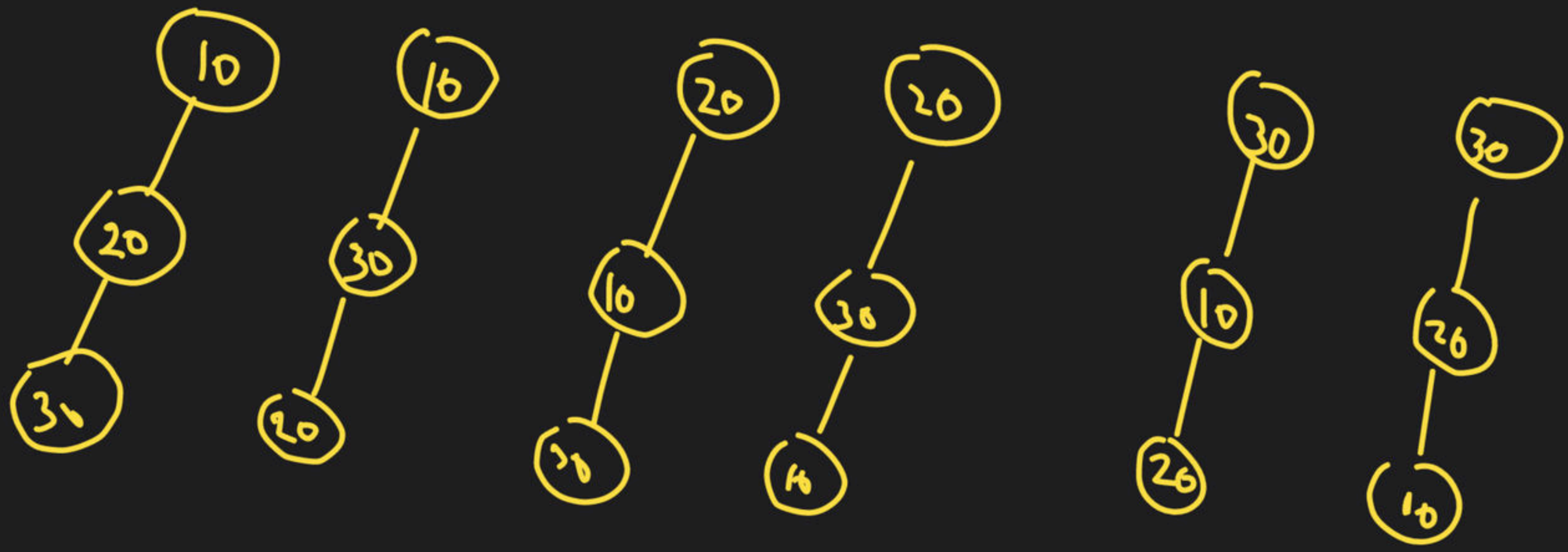


unacademy

3 distinct keys: 10, 20, 30

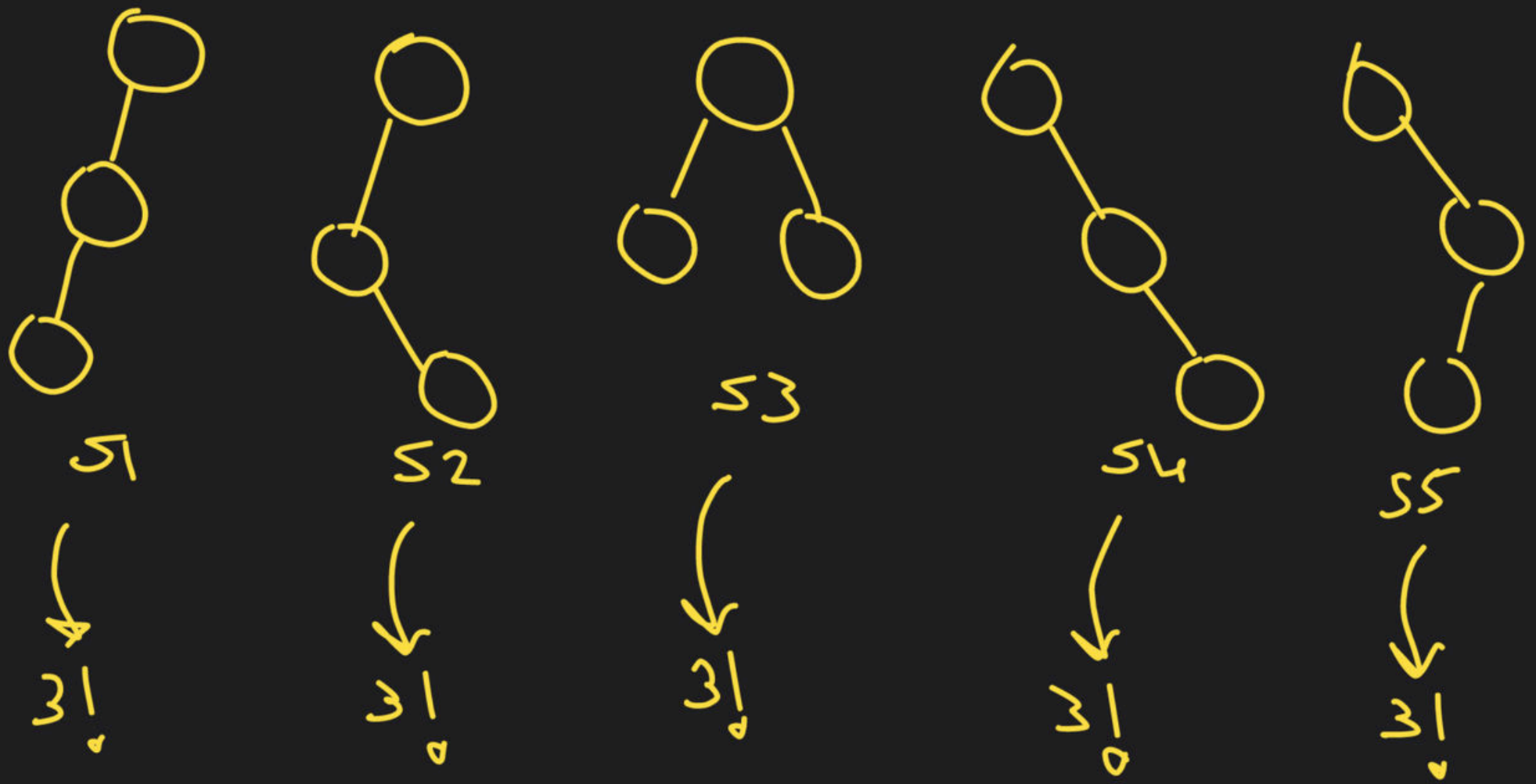
labelled binary tree with 3 distinct keys:

Sl



unacademy

$n=3 \Rightarrow 5$ unlabelled structure



labelled binary trees with 3 distinct keys

$$= \left(\begin{array}{c} \#\text{ of unlabelled binary} \\ \text{trees with 3 nodes} \end{array} \right) \times 3!$$

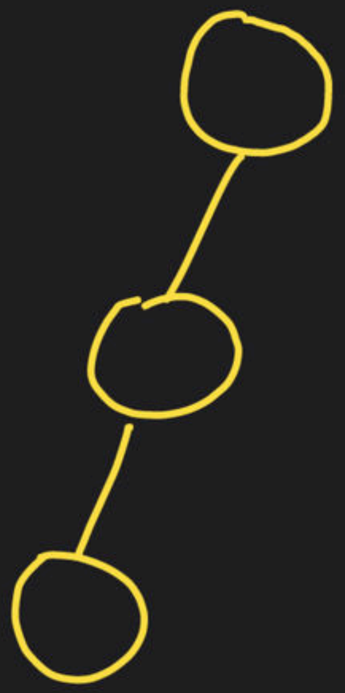
$$= 5 \times 6$$

$$= 30$$

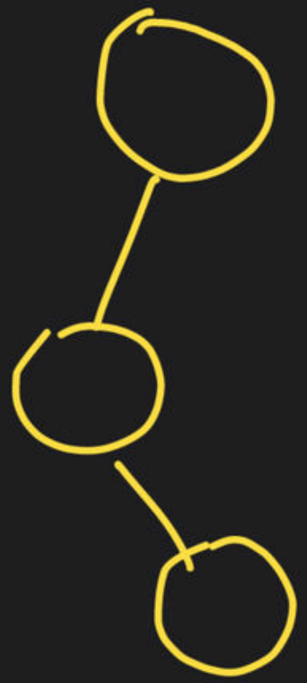
labelled binary trees with n distinct keys

$$= \left(\frac{{}^{2n}C_n}{n+1} \right) \times n!$$

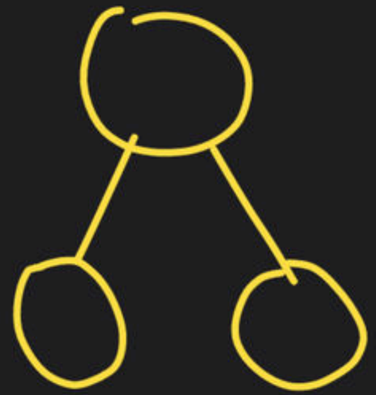
binary trees possible with preorder ABC



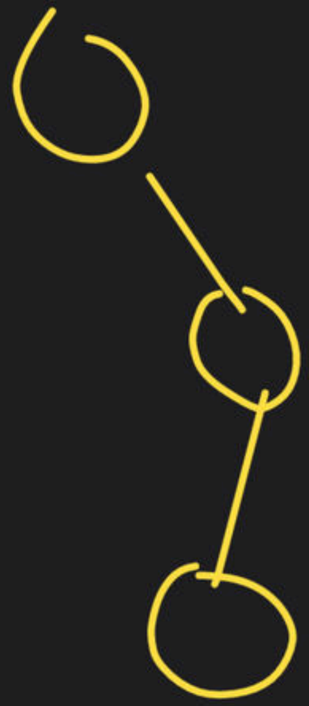
S1



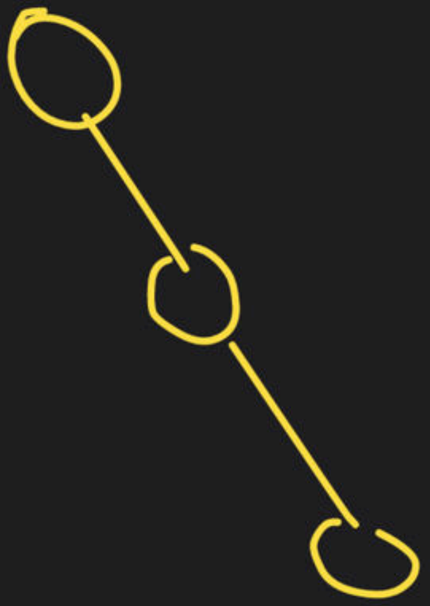
S2



S3



S4



S5

unacademy

For 1 structure ⇒ 1 such binary tree is possible

Pre: ABC

Pre: ACB

Y

Pre: BAC

X

Pre: BCA

X

Pre: CAB

X

Pre: CBA

X

binary trees possible with preorder ABC

= # binary tree structure with 3 nodes

binary trees with a given preorder of n length = $\frac{^{2n}C_n}{n+1}$

Pre: ABC

Given postorder (length n), no. of binary trees

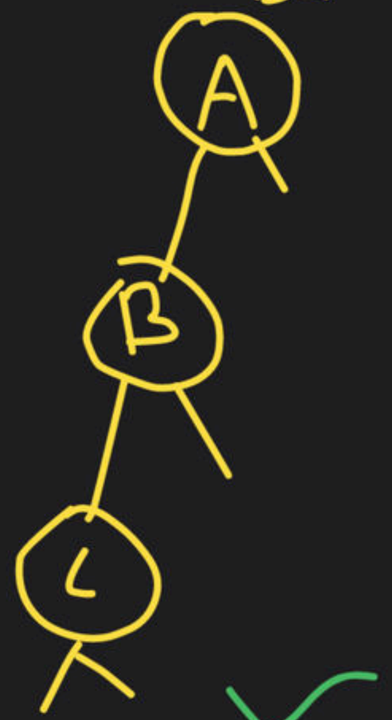
$$= \frac{2n C_n}{n+1}$$

With any 1 given traversal (Pre/In/Post), the no. of binary trees possible

$$= \frac{2n C_n}{n+1}$$

binary trees possible with preorder ABC and postorder CBA

Pre: ABC

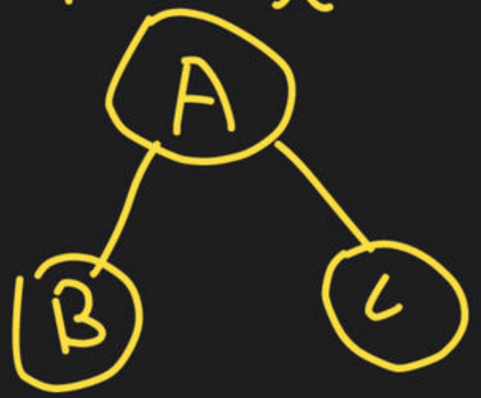


Post: CBA ✓

Pre: ABC

Post: CBA ✓

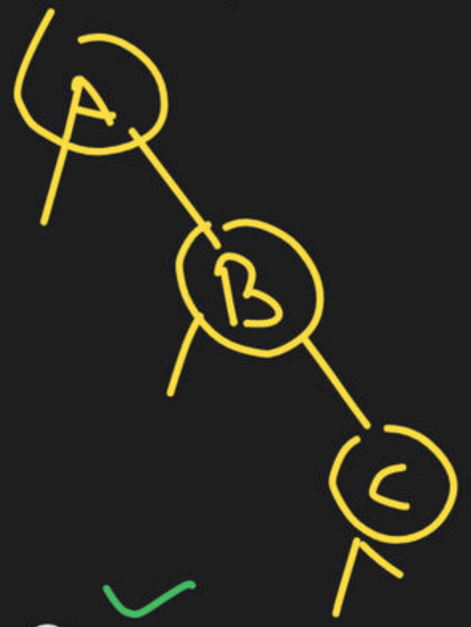
Pre: ABC



✗

Post: BCA

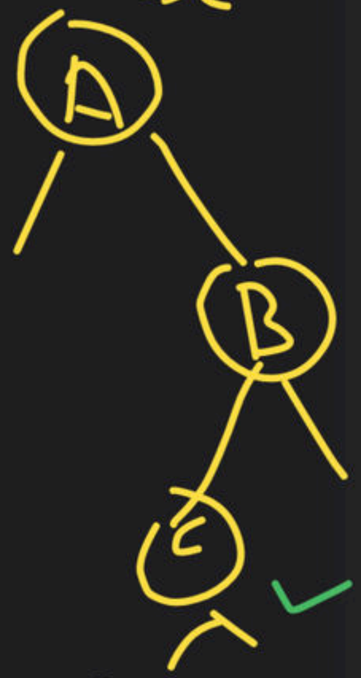
Pre: ABC



✓

Post: CBA

Pre: ABC



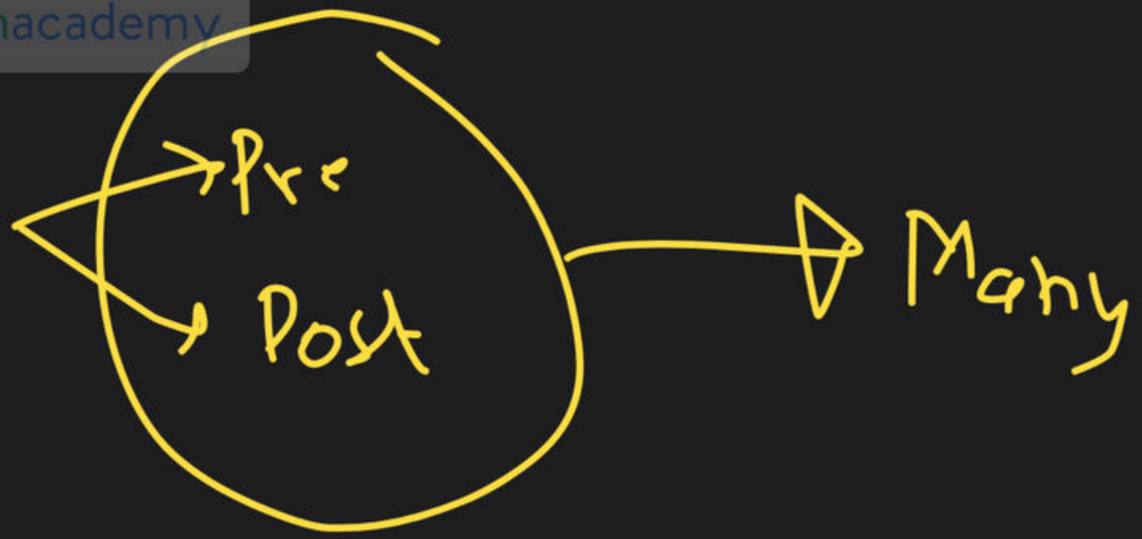
✓

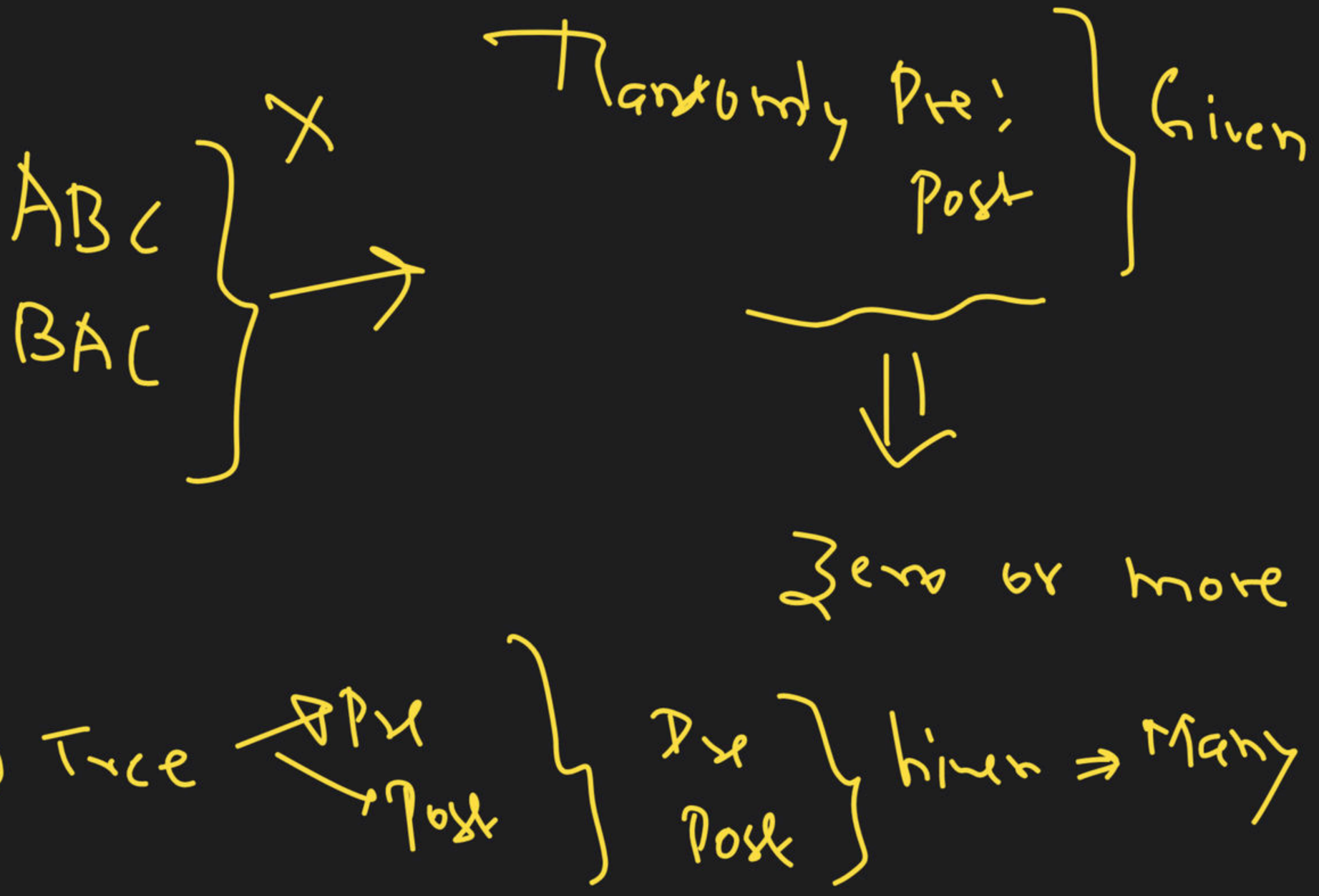
Post: CBA

unacademy

Pre:
Post
) → Many such trees

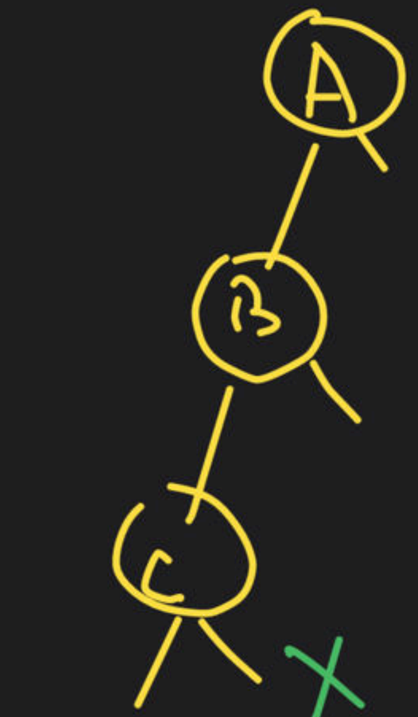
Pre: ABC
Post: BAC
) ⇒ No binary tree



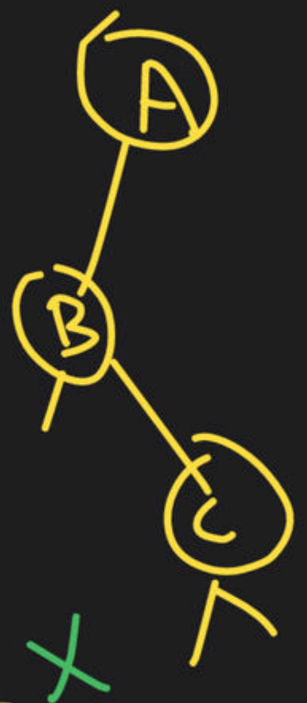

X
ABC
BAC
Randomly Pre:
Post
Given
Zero or more
Binary Tree
Pre
Post
Pre
Post
Given ⇒ Many

unacademy

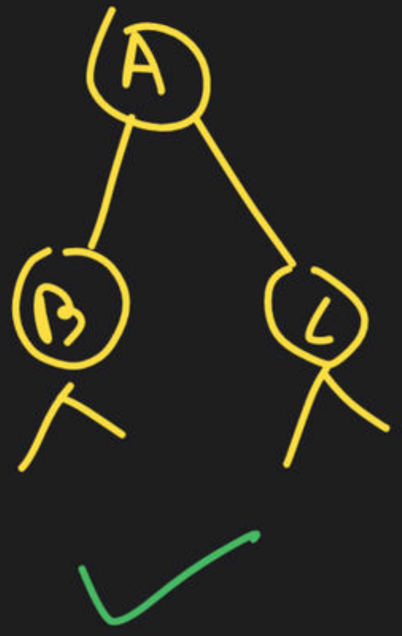
#binary trees with Preorder : ABC and Inorder : BAC



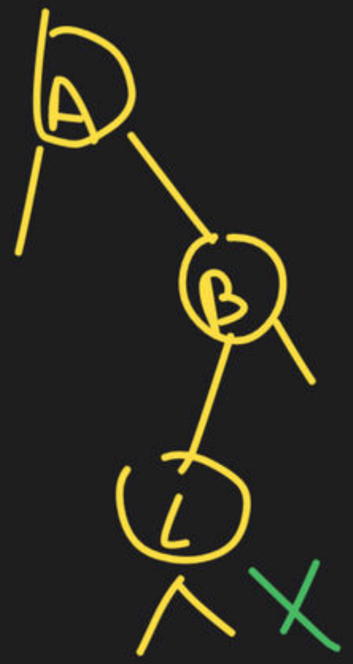
In: CBA



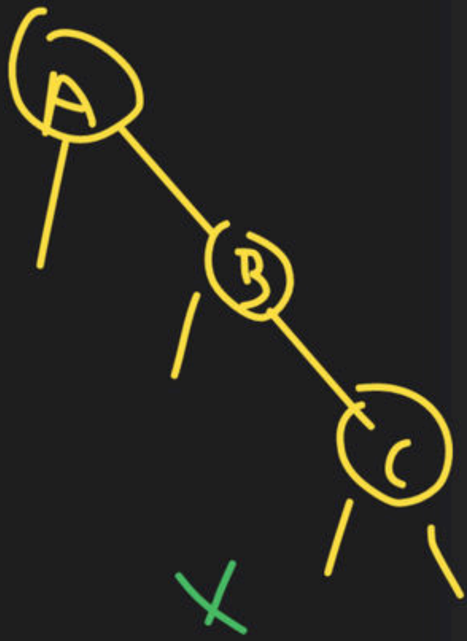
In: BCA



In: BAC



In: ACB



In: ABC

With a given Pre, In ⇒ Unique binary tree can be constd.

With a given Post, In ⇒ Unique binary tree can be constd.

unacademy

THANK YOU!

Here's to a cracking journey ahead!