



CS & IT ENGINEERING

Data Structure
& Programming

Tree

DPP - 01

Discussion Notes

By- Abhishek sir



#Q. The number of unlabeled binary trees possible with four nodes is 14.

Catalan No.

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

$$C_4 = \frac{1}{5} 8 C_4$$

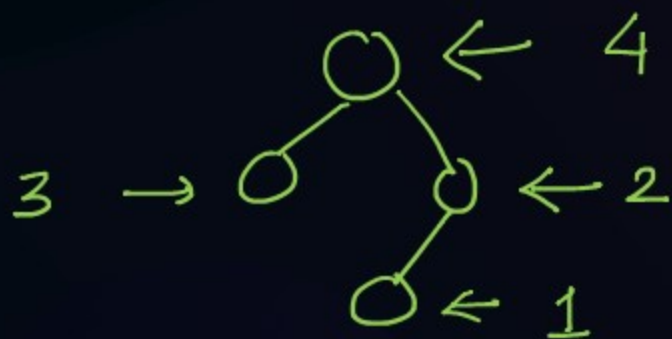
$$= \frac{1}{5} \frac{18}{18 \quad 14}$$

$$= \frac{1}{5} \frac{\cancel{8} \times 7 \times \cancel{6} \times 5^2}{4 \times \cancel{3} \times \cancel{2} \times 1} = 14$$

#Q. The number of labelled binary trees possible with the nodes-10, 30, 25, 40 is 336.

No. of unlabelled trees = 14

No. of ways we can label
= $4!$



No. of labelled trees = $14 \times 4!$
= $14 \times 24 = \underline{336}$

#Q. The number of binary search trees possible with the nodes-10, 30, 25, 40 is 14.



No. of Binary Search tree

one labelled tree can filled only

one way hence

$$\begin{aligned} \# \text{ BST} &= \# \text{ of unlabelled trees} \\ &= \underline{14} \end{aligned}$$

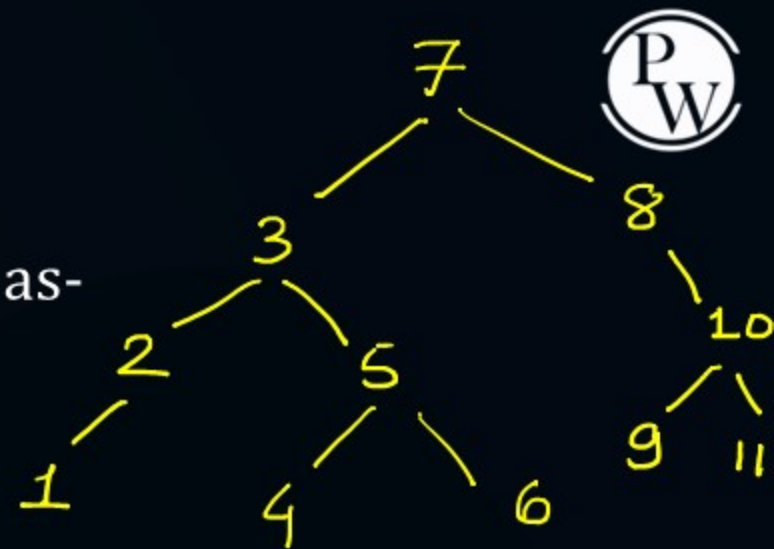
[MCQ]

Root Left Right

Left

#Q. The preorder traversal of a binary search tree is given as-
7, 3, 2, 1, 5, 4, 6, 8, 10, 9, 11

The post-order traversal of the above binary tree is-



A

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

B

1, 2, 4, 6, 5, 3, 9, 11, 10, 8, 7

C

1, 2, 4, 5, 6, 3, 9, 10, 11, 8, 7

D

11, 9, 10, 8, 6, 4, 5, 1, 2, 3, 7

only preorder

postorder : 1, 2, 4, 6, 5, 3

Answer B

[MCQ]



2, 1, 3
1 2 3

#Q. Consider the following two statements:

Statement P: The last elements in the pre-order and in-order traversal of a binary search tree are always same. *Incorrect*

Statement Q: The last elements in the pre-order and in-order traversal of a binary tree are always same. *Incorrect*

Which of the following tree is/are CORRECT?

A

Both P and Q only

B

Neither P nor Q

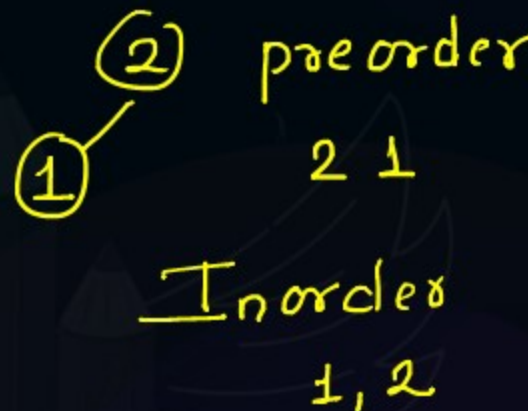
C

Q only

D

P only

BST - Binay Tree



Tree Node

#Q. Consider the following function:

```
struct treenode{
struct treenode *left; ✓
int data; ✓
struct treenode *right;
};
```

```
int func (struct treenode *t){
if(t==NULL) return 1; ✓
else if(t->left==NULL && t->right==NULL)
return 1;
```

```
else if
```

```
((t → left → data < t->data) && (t → right → data > t->data))
```

treenode pointer is Input.

(1) if t is NULL

Zero

1 element



Continue...

Recursion

(BST)

Tree structure



```
return func(t->left) && func(t->right);
```

```
else
```

```
return 0; ✓
```

```
}
```

↑ Logical operators && operator

Recursive

Assume t contains the address of the root node of a tree. The function-

✓ A

Returns 1 if the given tree is a Binary Search Tree.

Answer is A

B

Returns 0 if the given tree is a complete binary tree.

C

Returns 0 if the given tree is a Binary Search Tree.

D

Returns 1 if the given tree is a complete binary tree.

#Q. Consider the following function:

```
struct treenode{  
    struct treenode *left;  
    int data;  
    struct treenode *right;  
};
```

```
struct treenode * f(struct treenode *t, int x){  
    if(t==NULL) return NULL;  
    elseif(x==t->data) return ____a____;  
    else if (x<t->data) return ____b____;  
    else return ____c____;  
}
```

treenode structure

⑬

⑮ ←

Data found t

f(t → Left, x)

f(t → right, x)

Continue...

Searching algorithm

Root Node



[BST]

Assume t contains the address of the root node of a binary search tree. The function finds an element x in the BST and returns the address of the node if found.

Which of the following statement(s) is/are CORRECT?

Address of Node

A

a: NULL ; b: $f(t \rightarrow \text{left}, x)$; c: $f(t \rightarrow \text{right}, x)$

B

a: t ; b: $f(t \rightarrow \text{right}, x)$; c: $f(t \rightarrow \text{left}, x)$

~~D~~

a: NULL ; b: $f(t \rightarrow \text{right}, x)$; c: $f(t \rightarrow \text{left}, x)$

C

~~C~~

D

a: t ; b: $f(t \rightarrow \text{left}, x)$; c: $f(t \rightarrow \text{right}, x)$ ✓

Answer - D



THANK - YOU

