Branch : CSE & IT **Batch : Hinglish**

Data Structure

Tree

DPP 03

[NAT]

1. The number of unlabelled binary trees possible with four nodes is _________.

[NAT]

2. The number of labelled binary trees possible with the nodes-10, 30, 25, 40 is _________.

[NAT]

3. The number of binary search trees possible with the nodes-10, 30, 25, 40 is _________.

[MCQ]

4. The pre-order 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

[MCQ]

5. 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.

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

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

[MCQ]

6. 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))
return func(t->left) && func(t->right);
else
return 0;
}
```

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.

(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.

[MCQ]

7. 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______;
}
```

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?

(a) a: NULL ; b: f(t->left, x) ; c: f(t->right, x)

(b) a: t ; b: f(t->right, x) ; c: f(t->left, x)

(c) a: NULL ; b: f(t->right, x) ; c: f(t->left, x)

(d) a: t ; b: f(t->left, x) ; c: f(t->right, x)

Answer Key

1. (14)
2. (336)
3. (14)
4. (b)
5. (b)
6. (a)
7. (d)

Hint & Solutions

1. **(14)**

Number of unlabelled binary trees possible with 4 nodes

$$= \frac{1}{4+1} \times \frac{(2\times 4)!}{4!\ 4!}$$

$$= \frac{1}{5} \times \frac{8!}{4!\ 4!}$$

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

= 14

2. **(336)**

Number of labelled binary trees possible with 4 nodes-

= 4! × Number of unlabelled binary trees with 4 nodes

= 4! × 14

= 336

3. **(14)**

Number of BSTs with 4 = Number of unlabelled binary trees with nodes

4. **(b)**

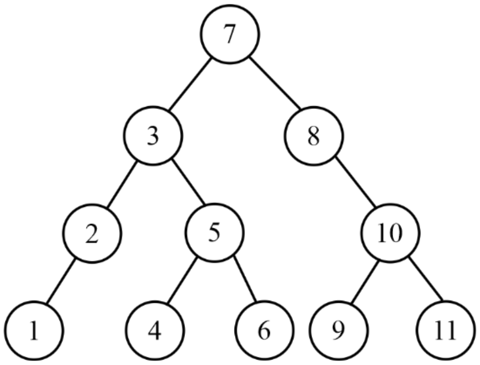
Pre-order traversal of BST:

7 3 2 1 5 4 6 8 10 9 11

In-order traversal of BST:

1 2 3 4 5 6 7 8 9 10 11

Tree is constructed as-



Post-order traversal-

1 2 4 6 5 3 9 11 10 8 7

5. **(b)**

P: INCORRECT. The last elements in the pre-order and in-order traversal of a binary search tree are not always same.(It violates for skewed BSTs)

Q: INCORRECT. The last elements in the pre-order and in-order traversal of a binary tree are not always same.

6. **(a)**

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

7. **(d)**

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