Data Structures & Programming

CS & IT

Tree

DPP: 1

Q1 The number of unlabeled binary trees possible with four nodes is _____.

Q2 The number of labelled binary trees possible with thenodes-10, 30, 25, 40 is _____.

Q3 The number of binary search trees possible with the nodes-10, 30, 25, 40 is _____.

Q4 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

Q5 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

Q6 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 binarytree.
(C) Returns 0 if the given tree is a Binary Search Tree.
(D) Returns 1 if the given tree is a complete binary tree.

Q7 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 abinary search tree. The function finds an element x inthe 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

Q1	14~14
Q2	336~336
Q3	14~14
Q4	(B)
Q5	(B)
Q6	(A)
Q7	(D)

Hints & Solutions

Q1 Text Solution:

Number of unlabelled binary trees possible with 4 nodes

$= \frac{1}{4+1} \times \frac{(2\times4)!}{4!4!}$

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

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

Q2 Text Solution:

Number of labelled binary trees possible with 4 nodes-

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

= 4! × 14

= 336

Q3 Text Solution:

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

Q4 Text Solution:

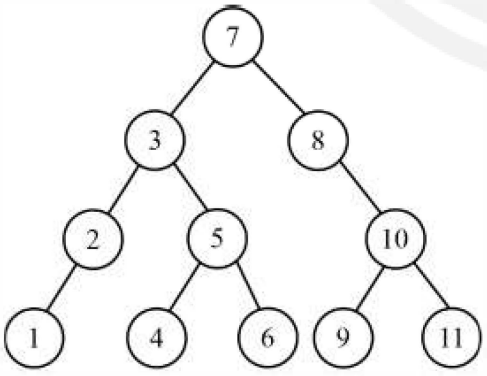
Pre-order traversal of BST:

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

Q5 Text Solution:

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.

Q6 Text Solution:

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

Q7 Text Solution:

```
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);
}
```