Branch : CSE/IT **Batch : English**

Data Structure & Programming

DPP

Tree

Section-01

[NAT]

1. A binary tree has 1024 leaves. The number of nodes in the tree having two children is _______________.

[MCQ]

2. The height of a tree is the length of the longest root-to-leaf path in it. The maximum and minimum number of nodes in a binary tree of height 9 are-

(a) 1024, 9
(b) 1023, 10
(c) 511, 9
(d) 512, 10

[NAT]

3. In a binary tree, the number of internal nodes of degree 1 is 6, and the number of internal nodes of degree 2 is 12. The number of leaf nodes in the binary tree is ________.

[MCQ]

4. A strict k-ary tree T is a tree that contains exactly 0 or k children. The number of leaf nodes in tree T if there are exactly 'p' internal nodes is-

(a) $(k - 1)p + 1$
(b) $pk + 1$
(c) $pk + 1 + p$
(d) None

[NAT]

5. A linked list is used to store a binary tree with 1024 nodes. The number of null pointers present is ___________.

[NAT]

6. Let T be a full binary tree with 4 leaves. (A full binary tree has every level full). Suppose two leaves x and y of T are chosen uniformly and independently at random. The expected value of the distance between x and y in T (i.e., the number of edges in the unique path between x and y) is (rounded off to 2 decimal places) _____.

[MCQ]

7. The number of leaf nodes in a rooted tree of n nodes, with each node having 0 or 2 children is-

(a) $\frac{n+1}{2}$
(b) $\frac{n-1}{2}$
(c) $\frac{n}{2}$
(d) $n - 1$

Answer Key

1. **(1023)**
2. **(b)**
3. **(13)**
4. **(a)**
5. **(1025)**
6. **(2.5)**
7. **(a)**

Hints and Solutions

1. (1023)

If there are 'n' leaf nodes, the number of internal nodes with 2 children is 'n – 1'.

2. (b)

Minimum number of nodes in a binary tree of height 9 = 10

Maximum number of nodes in a binary tree of height 9 = $2^{10} - 1 = 1023$

3. (13)

If there are 'n' internal nodes of degree 2, the number of leaf nodes is 'n + 1'.

4. (a)

Number of internal nodes	Number of leaf nodes
0	1
1	k
2	k + k – 1 i.e 2k – 1
3	2(2k – 1) – k i.e 3k – 2
. . .	
p	pk – (p – 1) i.e (k – 1)p + 1

5. (1025)

The number of null pointers = 1024 + 1 = 1025

6. (2.5)

Any two leaf nodes can be selected in 4 * 4 = 16 ways

Path length between x and y (i)	0	2	4
Number of ways	4	4	8
P(i)	$\frac{4}{16}$	$\frac{4}{16}$	$\frac{8}{16}$

The expected value of the distance between x and y in T-

$$E(i) = \sum i * P(i) = 0 \times \frac{4}{16} + 2 \times \frac{4}{16} + 4 \times \frac{8}{16} = 2.50$$

7. (a)

Number of nodes	Number of leaf nodes
1	1
3	2
5	3
7	4
. . .	. . .
n	$\frac{n+1}{2}$

Data Structure & Programming
Tree

Section-02

[MCQ]

1. Consider the following nested representation of binary trees: (X Y Z) indicates Y and Z are the left and right sub stress, respectively, of node X. Note that Y and Z may be NULL, or further nested. Which of the following represents a valid binary tree?
(a) (1 2 (4 5 6 7))
(b) (1 (2 3 4) 5 6) 7)
(c) (1 (2 3 4) (5 6 7))
(d) (1 (2 3 NULL) (4 5))

[MCQ]

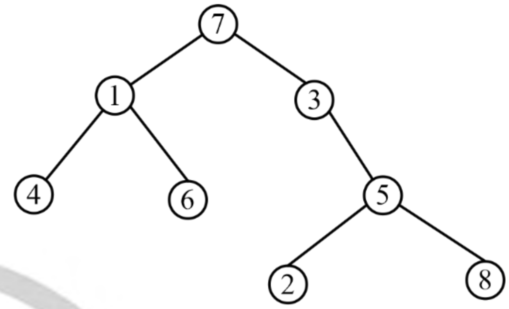
2. Consider the following two statements:
S1: It is possible to construct a binary tree uniquely whose post-order and pre-order traversals are given.
S2: It is possible to construct a binary tree uniquely whose in-order and pre-order traversals are given.
S3: It is possible to construct a binary tree uniquely whose post-order and level-order traversals are given.
Which of the following statement(s) IS/ARE INCORRECT?
(a) S1 only
(b) S2 only
(c) S1 and S3
(d) S3 only

[MCQ]

3. Let LASTPOST, LASTIN and LASTPRE denote the last vertex visited in a postorder, inorder and preorder traversal respectively, of a complete binary tree. Which of the following is always true?
(a) LASTIN = LASTPOST
(b) LASTIN = LASTPRE
(c) LASTPRE = LASTPOST
(d) None of the above

[MCQ]

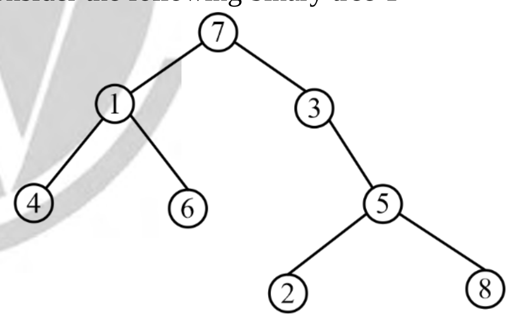
4. Consider the following binary tree T-



The in-order traversal of T is-
(a) 7 1 3 4 6 5 2 8
(b) 4 1 6 7 3 2 5 8
(c) 4 6 1 2 8 5 3 7
(d) 7 1 4 6 3 5 2 8

[MCQ]

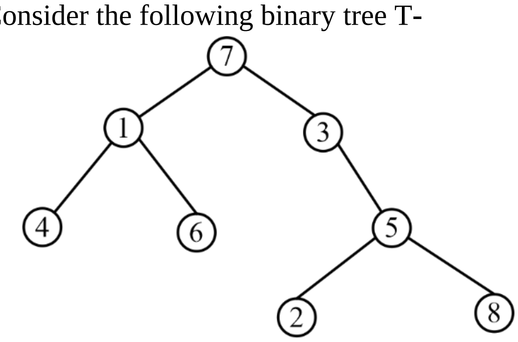
5. Consider the following binary tree T-



The pre-order traversal of T is-
(a) 7 1 3 4 6 5 2 8
(b) 4 1 6 7 3 2 5 8
(c) 4 6 1 2 8 5 3 7
(d) 7 1 4 6 3 5 2 8

[MCQ]

6. Consider the following binary tree T-



The post-order traversal of T is-

(a) 7 1 3 4 6 5 2 8
(b) 4 1 6 7 3 2 5 8
(c) 4 6 1 2 8 5 3 7
(d) 7 1 4 6 3 5 2 8

[NAT]

7. The pre-order traversal of a binary tree is 1, 2, 4, 7, 8, 3, 5, 6, 9. The in-order traversal of the same tree is 7 4 8 2 1 5 3 6 9. The height of a tree is the length of the longest path from the root to any leaf. The height of the binary tree above is _______.

[MCQ]

8. The post-order traversal of a binary tree is 9, 7, 4, 8, 2, 5, 1, 3, 6. The in-order traversal of the same tree is 9, 7, 8, 4, 5, 2, 6, 3, 1. The pre-order traversal of the above binary tree is-
(a) 1, 2, 4, 7, 9, 8, 5, 3, 6
(b) 1, 2, 4, 7, 8, 9, 5, 3, 6
(c) 1, 2, 3, 4, 5, 6, 7, 8, 9
(d) None of the above.

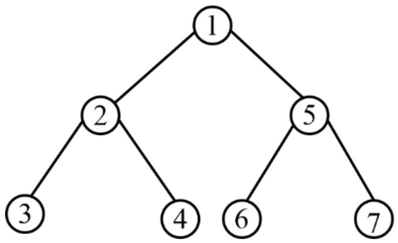
Answer Key

1. (c)
2. (c)
3. (d)
4. (b)
5. (d)
6. (c)
7. (3)
8. (a)

Hints and Solutions

1. (c)

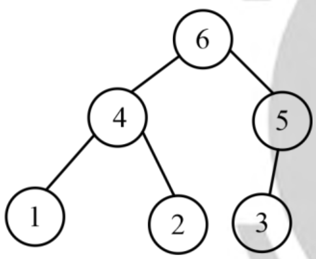
Correct



2. (c)

It is possible to construct a binary tree uniquely whose in-order and pre-order/post-order traversals are given.

3. (d)



In order: 1 4 2 6 3 5

Pre-order: 6 4 1 2 5 3

Post-order: 1 2 4 3 5 6

Clearly, LASTIN ≠ LASTPRE ≠ LASTPOST

4. (b)

The in-order traversal of T is- 4 1 6 7 3 2 5 8

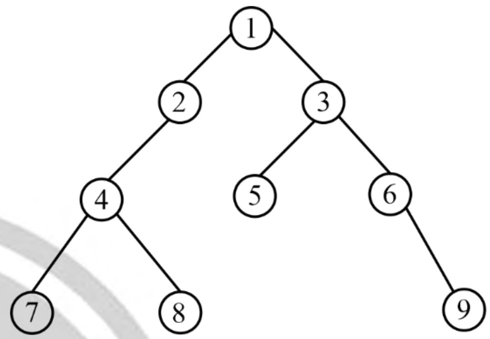
5. (d)

The pre-order traversal of T is- 7 1 4 6 3 5 2 8

6. (c)

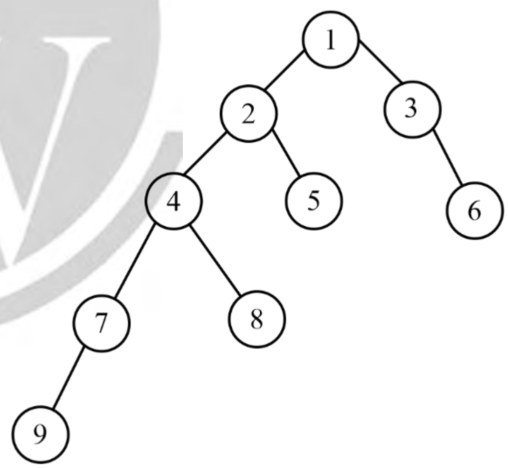
The post-order traversal of T is - 4 6 1 2 8 5 3 7

7. (3)



Height of the above binary tree = 3

8. (a)



The pre-order traversal of the above binary tree is- 1, 2, 4, 7, 9, 8, 5, 3, 6

Branch : CSE & IT **Batch : English**

Data Structure & Programming
Tree

Section 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)

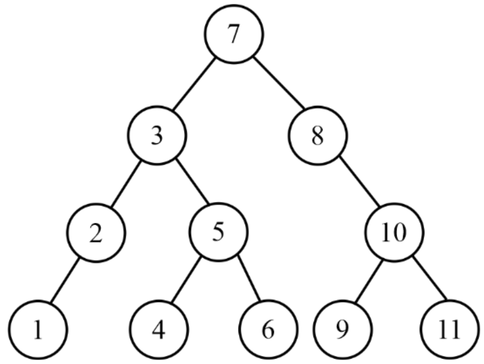
Pre-order traversal of BST:

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);
}
```

Data Structure & Programming

Tree

Section-04

[MCQ]

1. Consider the following function:

```
struct treenode
{
    struct treenode *left;
    int data;
    struct treenode *right;
};
int func(struct treenode *p, struct treenode *q){
if(p==NULL && q==NULL) return 1;
if((!p && q) || (!q && p)) return 0;
return (p->data==q->data) && func(p->left, q->right)
&& func(p->right, q->left);
}
```

Initially the addresses of root node of two trees are passed into p and q respectively, the function-

(a) Returns 1 iff the two trees are identical.
(b) Returns 1 iff the two trees are mirror images of each other.
(c) Returns 1 iff the two trees emerge from the same root node.
(d) None of the above.

[MCQ]

2. Consider the following function:

```
struct treenode
{
    struct treenode *left;
    int data;
    struct treenode *right;
};
int func(struct treenode *p, struct treenode *q){
if(p==NULL && q==NULL) return 1;
if((!p && q) || (!q && p)) return 0;
return (p->data==q->data) && func(p->left, q->left)
&& func( p->right, q->right);
}
```

Initially the addresses of root node of two trees are passed into p and q respectively, the function-

(a) Returns 1 iff the two trees are identical.
(b) Returns 1 iff the two trees are mirror images of each other.
(c) Returns 1 iff the two trees emerge from the same root node.
(d) None of the above

[MCQ]

3. Consider the following function:

```
struct treenode
{
    struct treenode *left;
    int data;
    struct treenode *right;
};
int func(struct treenode *p)
{
    if(p==NULL) return 1;
    else if(p->right!=NULL) return 0;
    return func(p->left);
}
```

Initially p contains the root node address of the tree, the function-

(a) Returns 1 if a binary tree is left-skewed.
(b) Returns 1 if a binary tree is right-skewed.
(c) Returns 1 if a binary tree is not right-skewed.
(d) None of the above.

[MCQ]

4. Consider the following functions:

```
struct treenode
{
    struct treenode *left;
    int data;
    struct treenode *right;
};
int f1(struct treenode *t)
{
    if(t==NULL) return 1;
    else if(t->left!=NULL) return 0;
```

```
    return func(t->right);
}
int * f2 (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;
}
int f3(){return f2(t) && f1(t);}
```

Assume, t is a pointer to the root node of a binary tree, the function f(3):

(a) Returns 1 if the binary tree is a left-skewed BST
(b) Returns 1 if the binary tree is not a left-skewed BST
(c) Returns 1 if the binary tree is a right-skewed BST
(d) None of the above.

[MCQ]

5. Consider the following function:

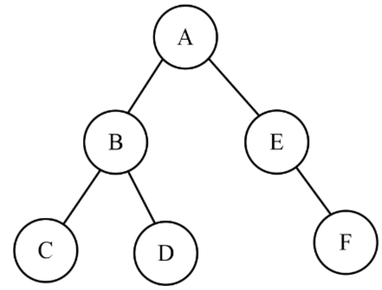
```
struct treenode
{
    struct treenode *left;
    int data;
    struct treenode *right;
};
int func(struct treenode *t)
{
    if(t==NULL) return 0;
    elseif(t->left==NULL && t->right==NULL)
    return 1;
    else
    return 1+func(t->left)+func(t->right);
}
```

Assume, t is a pointer to the root node of a binary tree, the function computes-

(a) Number of leaf nodes in the binary tree
(b) Number of internal nodes in the binary tree
(c) Total number of nodes in the binary tree
(d) None of the above

[MCQ]

6. The given tree is passed to the following function:




```
void func(struct treenode *t)
{
  if(t)
  {
    printf("%d", t->data);
    func(t->right);
    printf("%d", t->data);
    func(t->left);
  }
}
```

The output string is-

(a) AEFFEBDDCCBA
(b) AEFFEABDDBCC
(c) AEFFEBDDCCBA
(d) None of the above

[MCQ]

7. Consider the following function:

```
struct treenode
{
    struct treenode *left;
    int data;
    struct treenode *right;
};
void func(struct treenode *p){
while(p->left!=NULL) p=p->left;
printf("%d", p->data);
}
```

If the address of the root node of the BST is passed to p, the above function prints-
(Assume, the tree contains at least one node)

(a) The maximum element in the BST
(b) The ancestor of two leftmost leaf nodes
(c) The minimum element in BST
(d) None of the above

[MCQ]

8. Consider the following two statements:

P: The minimum number of nodes in a complete binary tree is 2^{h+1}.

Q: A binary search tree is always a complete binary tree.

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

(a) P only (b) Q only

(c) Both P and Q (d) Neither P nor Q

Answer Key

1. (b)
2. (a)
3. (a)
4. (c)
5. (c)
6. (b)
7. (c)
8. (d)

Hints and Solutions

1. **(b)**
The function returns 1 iff the two trees are mirror images of each other.

2. **(a)**
The function returns 1 iff the two trees are identical each other.

3. **(a)**
The function returns 1 iff the binary tree is left-skewed.

4. **(c)**
The function returns 1 iff the binary tree is right-skewed BST.

5. **(c)**
The function computes the total number of nodes in a binary tree.

6. **(b)**
The output string is "AEFFEABDDBCC".

7. **(c)**
The function returns the minimum element in a binary search tree.

8. **(b)**

P: INCORRECT. The minimum number of nodes in a complete binary tree is 2^h.
Q: INCORRECT.A binary search tree is may not be a complete binary tree.

Branch : CSE & IT **Batch : English**

Data Structure & Programming
Tree

Section 05

[NAT]

1. The minimum number of nodes in AVL tree of height 6 is _______.

(Assume that the height of the root node is 1)

[MCQ]

2. Consider the following statements:

P: An AVL tree is a height-balanced complete binary tree.

Q: A heap is necessarily a complete binary tree.

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

(a) P only
(b) Q only
(c) Both P and Q
(d) Neither P nor Q

[NAT]

3. The total number of ways in which a max-heap can be constructed with the keys-7, 6, 1, 4, 5, 2, 3 is ________.

[MCQ]

4. Consider the following statements:

P: If the root node of a BST is deleted, it can be replaced by inorder predecessor.

Q: If the root node of a BST is deleted, it can be replaced by preorder successor.

Which of the following is/are CORRECT?

(a) P only
(b) Q only
(c) Both P and Q
(d) Neither P nor Q

[MSQ]

5. Consider the following operations in a BST-

INSERT(23), INSERT(17), INSERT(25), INSERT(4), INSERT(21), INSERT(1), INSERT(7), DELETE(17), DELETE(23).

The post-order traversal of the resultant BST is-

(a) 1, 7, 4, 21, 25
(b) 1, 4, 7, 25, 21
(c) 1, 4, 21, 7, 25
(d) None of the above

[MSQ]

6. Which of the following sequence(s) of array form a heap?

(a) 23, 17, 14, 6, 13, 10, 1, 12, 7, 5
(b) 1, 5, 10, 6, 7, 12, 13, 14, 17, 23
(c) 23, 17, 14, 7, 13, 10, 1, 5, 6, 12
(d) 1, 5, 10, 12, 13, 7, 14, 17, 23, 6

[NAT]

7. Consider the following statements:

P: The accepted balanced factor in an AVL tree are –1, 0 and +1.

Q: The height of an AVL tree with n nodes is given as $\lceil \log_2 n \rceil$.

The number of INCORRECT statements is ______.

[NAT]

8. Construct an AVL tree with the following keys:

12, 10, 15, 14, 13, 17, 8

The immediate left child key value of the root node of the AVL tree is __________.

Answer Key

1. (20)
2. (b)
3. (80)
4. (a)
5. (a, b, c)
6. (b, c)
7. (0)
8. (12)

Hint & Solutions

1. **(20)**

The minimum number of nodes in an AVL tree of height 'h' is given by-

$n(h) = n(h-1) + n(h-2) + 1$

$n(1) = 1, n(2) = 2, n(3) = 4, n(4) = 7, n(5) = 12,$
$n(6) = 20$

2. **(b)**

P: INCORRECT. An AVL tree is not necessarily a complete binary tree.

Q: CORRECT. A heap is necessarily a complete binary tree.

3. **(80)**

$T(n) = 1 * \binom{n-1}{k} * T(k) * T(n-k-1)$

Here n = 7, k = 3

$T(7) = 1 * \binom{6}{3} * T(3) * T(3)$

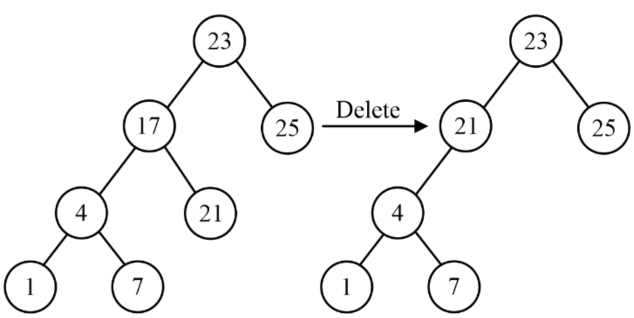
Now, T(3) = 2

$T(7) = 1 * \binom{6}{3} * 2 * 2 = 80$

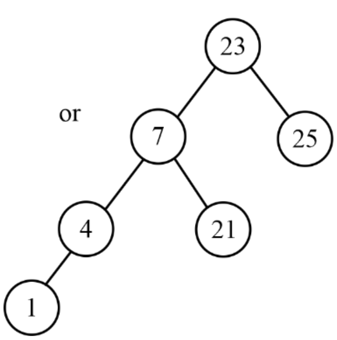
4. **(a)**

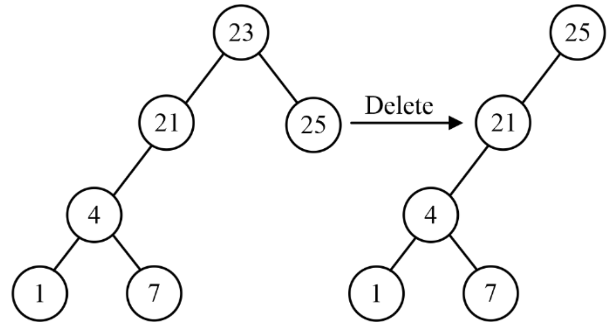
If the root node of a BST is deleted, it can be replaced by inorder predecessor/successor.

5. **(a, b, c)**

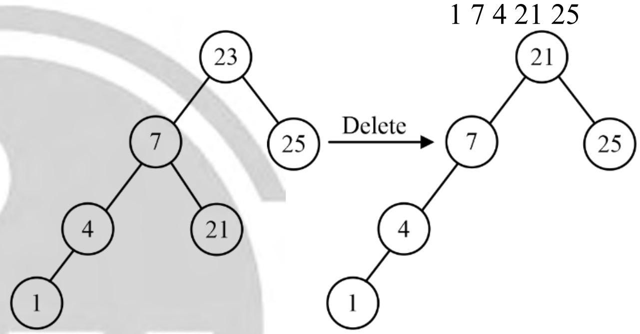


or



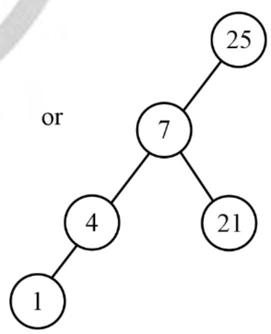


Post order traversal: -
1 7 4 21 25



Post order traversal: -
1 4 7 25 21

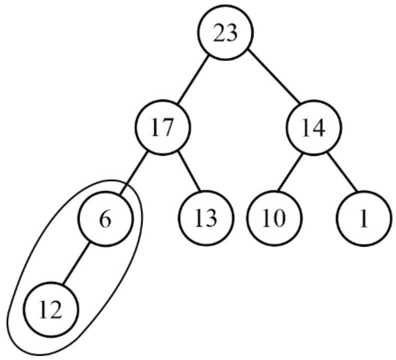
or



Post order traversal: -
1 4 21 7 25

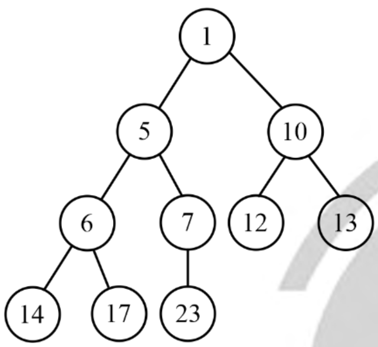
6. **(b, c)**

(a)



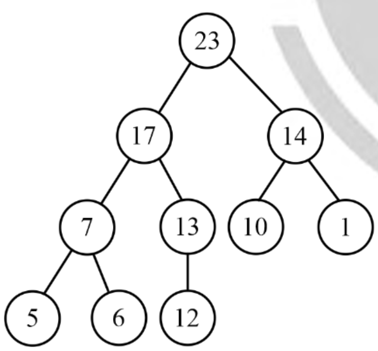
Not possible defies max-heap property

(b)



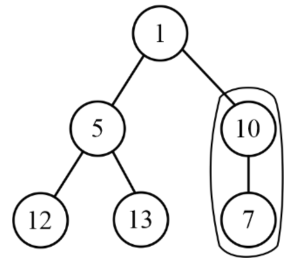
Satisfies min-heap property

(c)



Satisfies max-heap property.

(d)



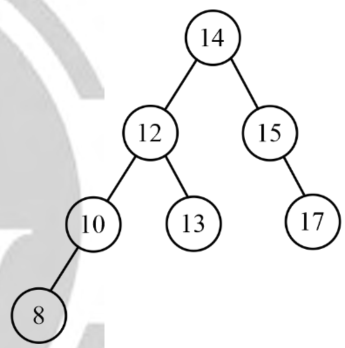
Not possible defies min-heap property

7. **(0)**

Both the statements are CORRECT.

8. **(12)**

Resultant AVL tree:



Branch : CSE/IT Batch : English

Data Structure & Programming
Tree

Section-06

[NAT]

1. The maximum number of comparisons to find the maximum element in a min heap of 1024 elements is ____________

[MCQ]

2. Consider the array given below:

50	40	10	5	60	70	40	15	80

The minimum number of comparisons required to convert the above array into max heap is _______

[NAT]

3. Consider the array given below:

50	40	10	5	60	70	40	15		80

The minimum number of swap operations required to convert the above array into max-heap is ________.

[MCQ]

4. Consider the array given below:

50	40	10	5	60	70	40	15	80

The resultant max-heap using bottom-up approach of build heap is-

(a) 80, 60, 70, 40, 50, 10, 40, 15, 5
(b) 80, 70, 60, 50, 40, 10, 40, 5, 15
(c) 80, 70, 60, 50, 40, 40, 15, 10, 5
(d) None of the above

[NAT]

5. Consider a sequence of elements are inserted into a max-heap one after another as-
50, 40, 10, 5, 60, 70, 40, 15, 80
The number of shift operations required in building the heap one element at a time is _____________.

[MCQ]

6. Consider a sequence of elements are inserted into a max-heap one after another as-
50, 40, 10, 5, 60, 70, 40, 15, 80
The resultant max-heap using bottom-up approach of build heap is-

(a) 80, 60, 70, 40, 50, 10, 40, 15, 5
(b) 80, 70, 60, 50, 40, 10, 40, 5, 15
(c) 80, 70, 60, 50, 40, 40, 15, 10, 5
(d) None of the above

[MCQ]

7. Consider the following two statements:

P: The number of comparisons required to find the minimum element in a min heap of n elements is n-1.

Q: Only one comparison is required to find the minimum element in a max heap of n elements.

Which of the following is/are CORRECT?

(a) P only
(b) Q only
(c) Both P and Q
(d) Neither P nor Q

Answer Key

1. (511)
2. (10)
3. (5)
4. (a)
5. (8)
6. (b)
7. (d)

Hints and Solutions

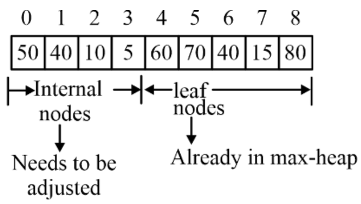
1. **(511)**

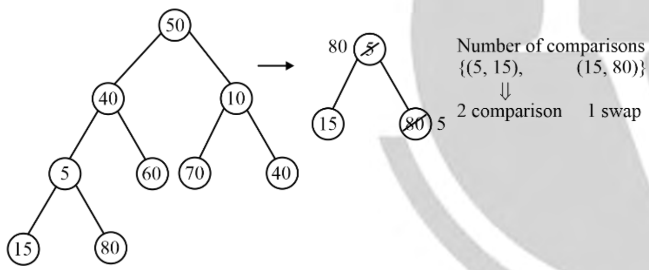
The maximum element will be present in the leaf nodes.

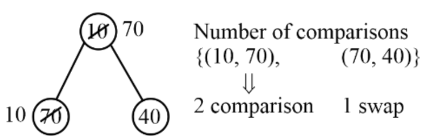
Number of leaf nodes in the min-heap of 1024 elements = 1024/2 = 512

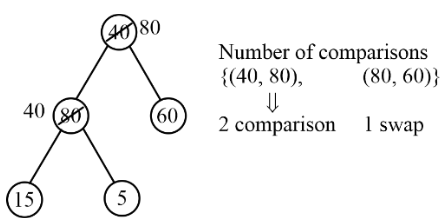
Maximum number of comparisons to find the maximum element in a min heap of 1024 elements = 512 – 1 = 511.

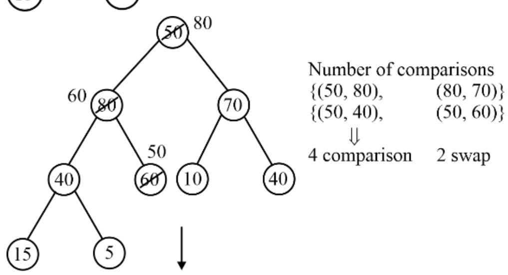
2. **(10)**

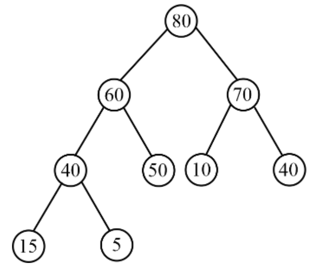




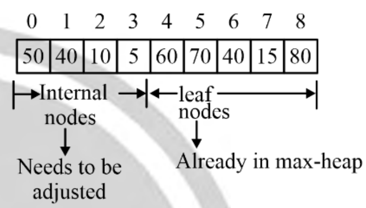


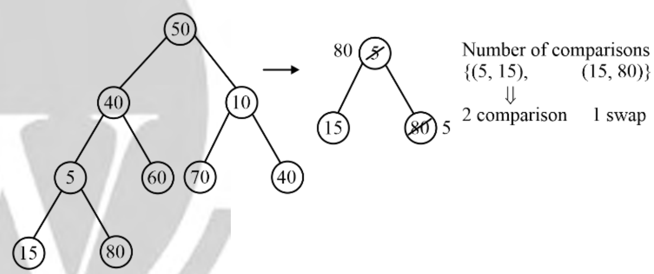


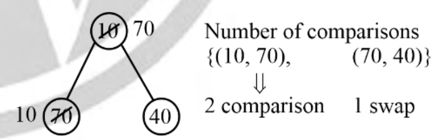


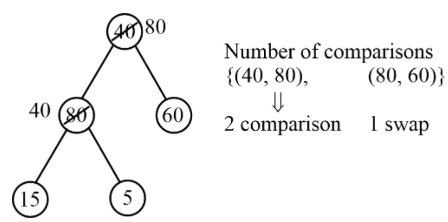


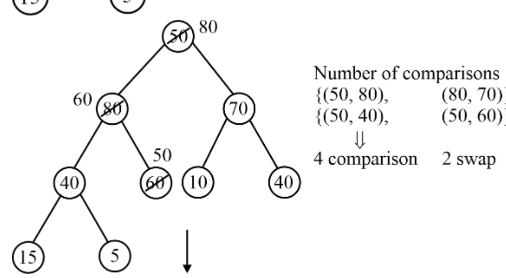
3. **(5)**

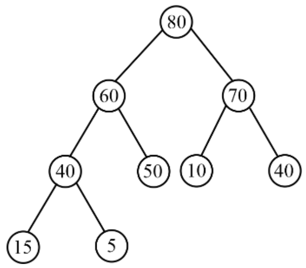




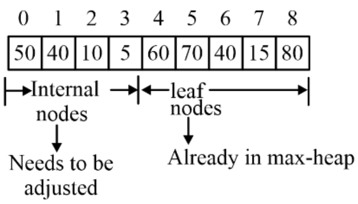


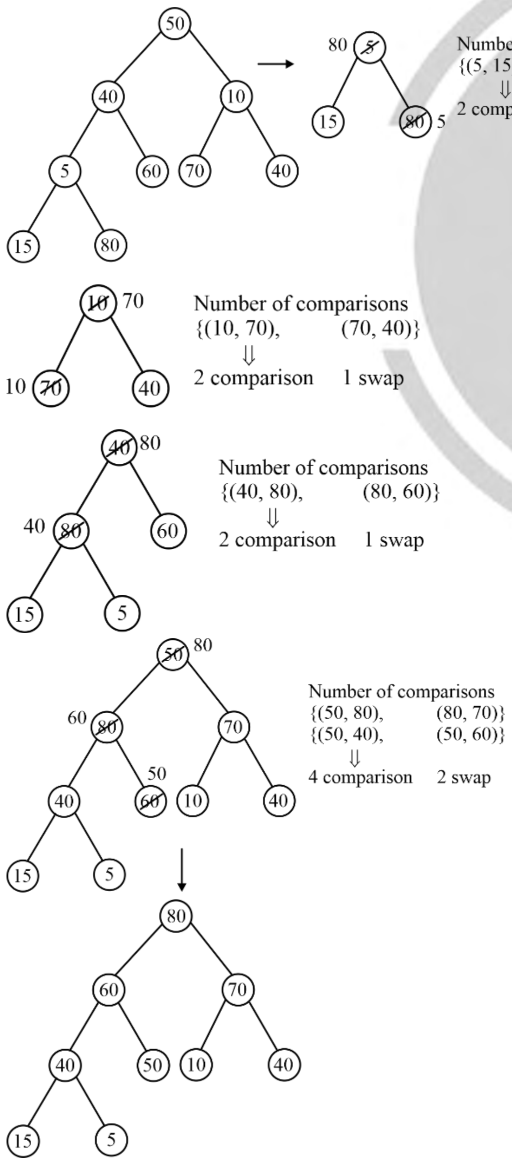




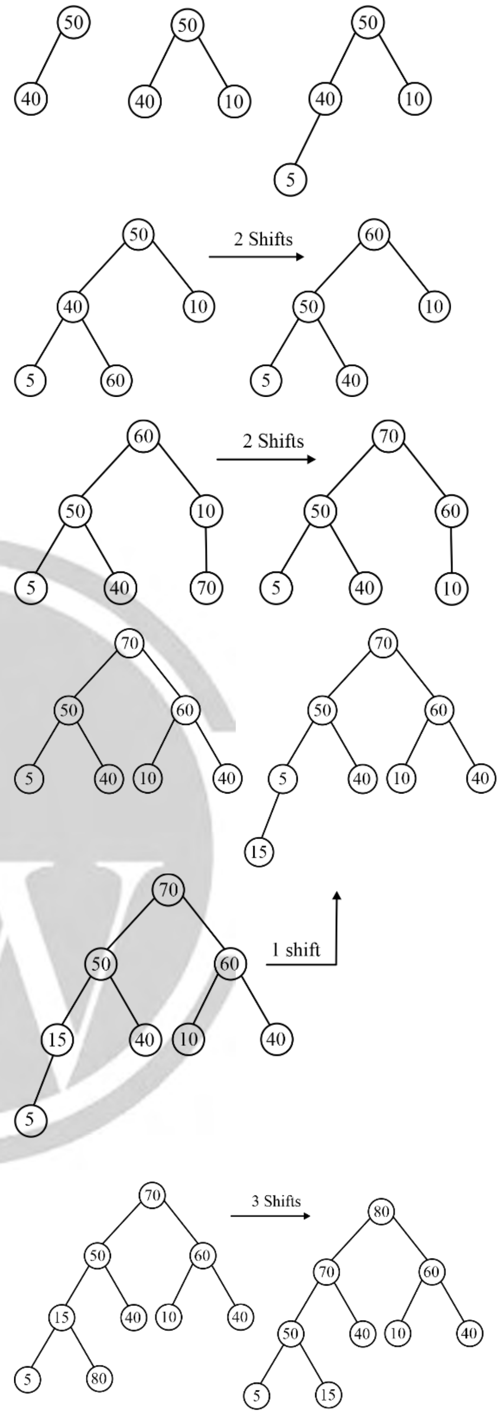


4. (a)





5. (8)



6. (b)

The resultant max-heap is-

80, 70, 60, 50, 40, 10, 40, 5, 15

7. (d)

P: INCORRECT. The number of comparisons required to find the minimum element in a min heap of n elements is 1.

Q: INCORRECT. Only one comparison is not sufficient to find the minimum element in a max heap of n elements.

Branch : CSE/IT Batch : English

Data Structure & Programming
Tree

Section-07

[MCQ]

1. Which of the following is/are correct inorder traversal sequence(s) of binary search tree(s)?

I. 3, 5, 7, 8, 15, 19, 25
II. 5, 8, 9, 12, 10, 15, 25
III. 2, 7, 10, 8, 14, 16, 20
IV. 4, 6, 7, 9 18, 20, 25

(a) I and IV (b) II and III
(c) II and IV (d) II only

[MCQ]

2. What is the worst-case time complexity of inserting n^2 elements into an AVL-tree with n elements initially?

(a) $O(n^2)$ (b) $O(n^2 \log n)$
(c) $O(n^4)$ (d) $O(n^3)$

[MCQ]

3. Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the pre-order traversal sequence of the resultant tree?

(a) 7 5 1 0 3 2 4 6 8 9
(b) 0 2 4 3 1 6 5 9 8 7
(c) 0 1 2 3 4 5 6 7 8 9
(d) 9 8 6 4 2 3 0 1 5 7

[MCQ]

4. Consider the following statements.

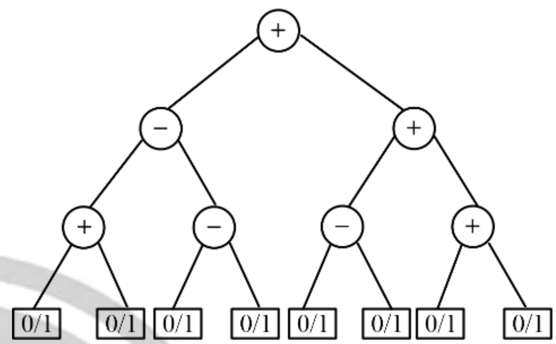
S_1: The sequence of procedure calls corresponds to a preorder traversal of the activation tree.
S_2: The sequence of procedure returns corresponds to a postorder traversal of the activation tree.

Which one of the following options is correct?

(a) S_1 only (b) S_2 only
(c) Both S_1 and S_2 (d) Neither S_1 nor S_2

[NAT]

5. Consider the expression tree shown. Each leaf represents a numerical value, which can either be 0 or 1. Over all possible choices of the values at the leaves, the maximum possible value of the expression represented by the tree is ___.



[MCQ]

6. A Binary Search Tree (BST) stores values in the range 37 to 573. Consider the following sequence of keys.

I. 81, 537, 102, 439, 285, 376, 305
II. 52, 97, 121, 195, 242, 381, 472
III. 142, 248, 520, 386, 345, 270, 307
IV. 550, 149, 507, 395, 463, 402, 270

Suppose the BST has been unsuccessfully searched for key 273. Which all of the above sequences list nodes in the order in which we could have encountered them in the search?

(a) I and III (b) II and III
(c) III and IV (d) III only

[NAT]

7. A complete n-ary tree is a tree in which each node has n children or no children. Let I be the number of internal nodes and L be the number of leaves in a complete n-ary tree. If L = 41, and I = 10, what is the value of n? _____________.

[MCQ]

8. A Priority-Queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is given below: 10, 8, 5, 3, 2. Two new elements '1' and '7' are inserted in the heap in that order. The level-order traversal of the heap after the insertion of the elements is:

(a) 10, 8, 7, 3, 2, 1, 5
(b) 10, 8, 7, 2, 3, 1, 5
(c) 10, 8, 7, 3, 2, 5, 1
(d) None of the above

Answer Key

1. **(a)**
2. **(b)**
3. **(a)**
4. **(c)**
5. **(6)**
6. **(d)**
7. **(5)**
8. **(a)**

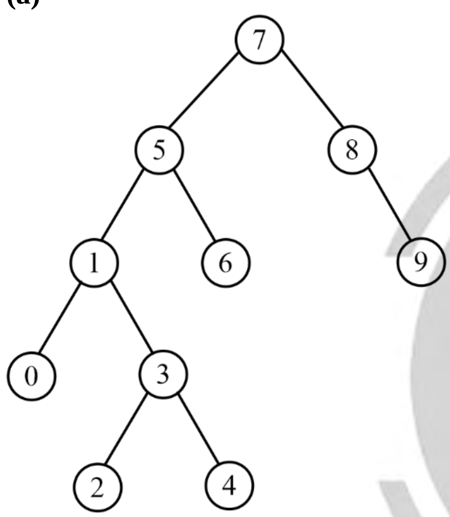
Hints and Solutions

1. (a)
The inorder traversal of BST is always in sorted order.

2. (b)

The worst-case time complexity of inserting n^2 elements into an AVL-tree with n elements initially is $O(n^2 \log n)$

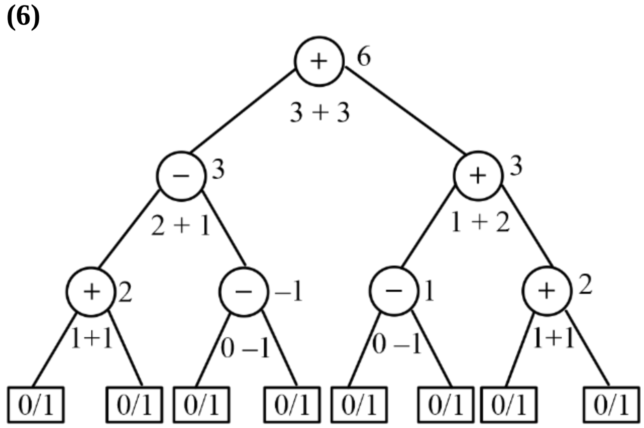
3. (a)



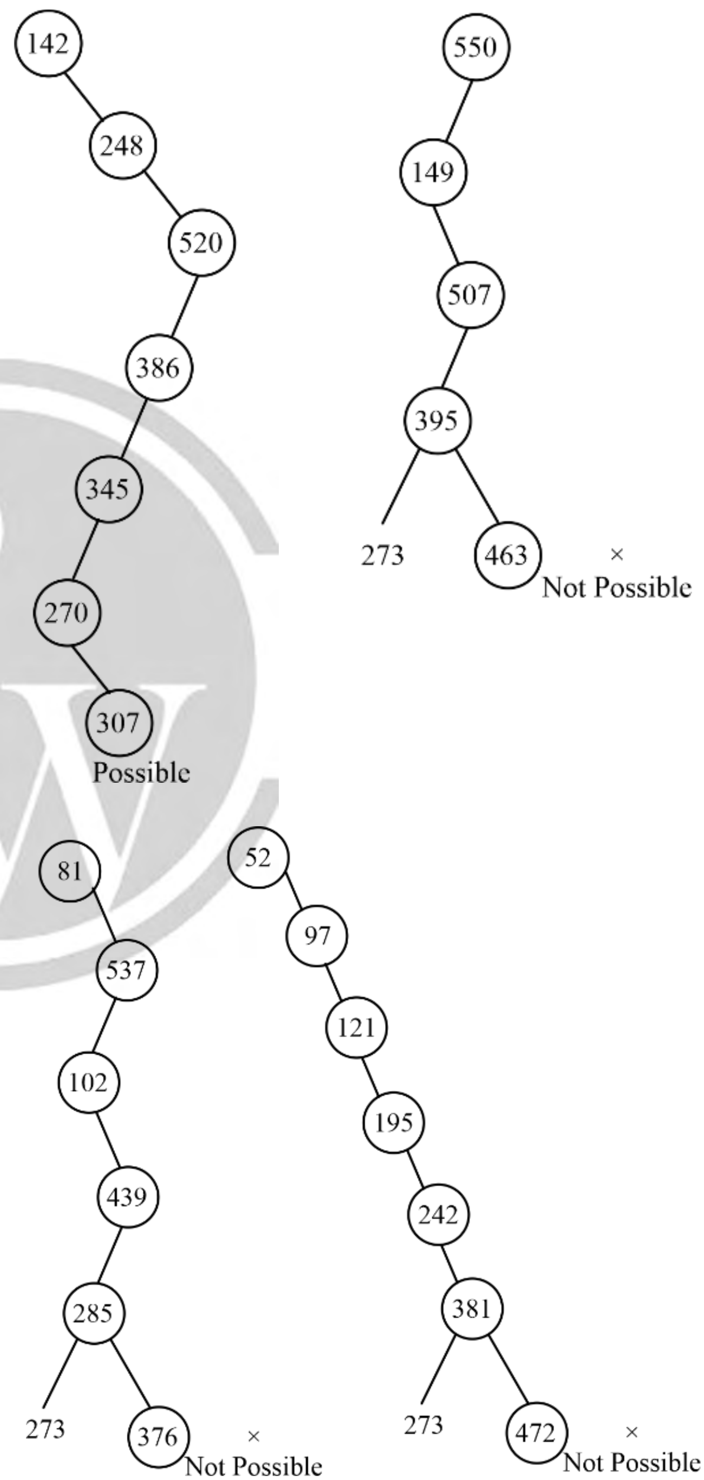
Preorder traversal
7 5 1 0 3 2 4 6 8 9

4. (c)
Both S_1 and S_2 are CORRECT.

5. (6)



6. (d)



7. (5)

$L = (N - 1) * I + 1$

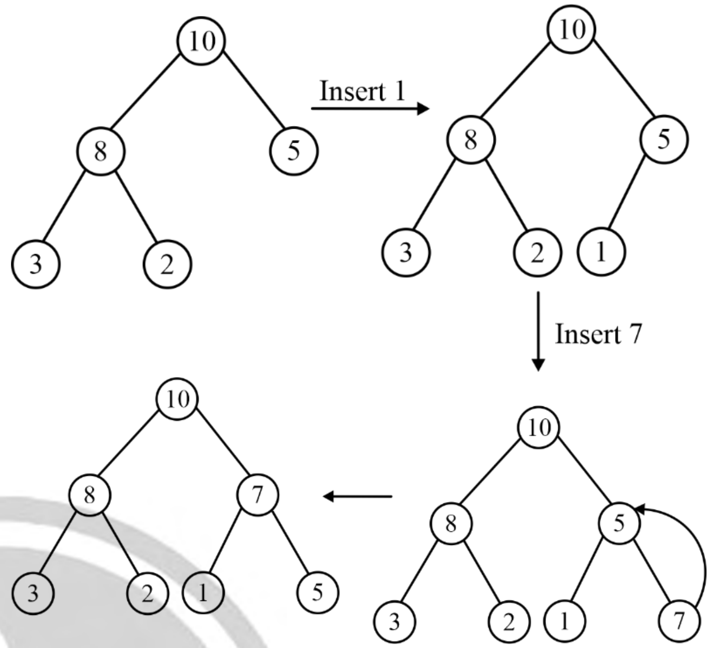
$L = 41, \; I = 10$ –

$41 = (N - 1) * 10 + 1$

$$N - 1 = \frac{40}{10} = 4$$

$N = 4 + 1 = 5$

8. (a)



Level order Traversal:

10 8 7 3 2 1 5