BRANCH:CSE/IT Batch-Hinglish

Data structure &Programming
Arrays

DPP-02

[NAT]

1. Consider a lower triangular 2D array arr[][5] with 15 elements. The number of rows in arr is- ________

[NAT]

2.
Consider an integer 2D array a[-7 to +7] [-7 to +7] that stores an upper triangular matrix uppertm where uppertm[i][j] is 1 for all i>=j. The sum of all the elements in the array is _____________.

[NAT]

3. Consider an integer lower triangular 2D array arr[-16 to +15][-16 to +15] having base address 1000. If the size of the integer is 4 bytes, the address of the element arr[8][7]is- _____________

[NAT]

4. Consider an integer upper triangular 2D array arr[-8 to +7][-8 to +7] having base address 1000. If the size of integer is 4 bytes, the address of the element present at location arr[-6][4] is- _____________.

[NAT]

5. Consider the natural numbers starting from 1 are stored in a lower triangular matrix arr[-3 to 3][-3 to 3]. Find the element present at location arr[1][2]. ________________

[NAT]

6. Consider the natural numbers starting from 1 are stored in a upper triangular 2D array arr[-3 to 3][-3 to 3]. Find the element present at location arr[1][2]. _________________.

[NAT]

7. Consider a 2D array arr[-4 to +4][-4 to 4] stores an upper triangular matrix. Find the address of the location arr[-1][-2] if the starting address of the array is 500 and size of each element is 8 bytes. Assume that elements are stored in column-major order. ________________

[NAT]

8. Consider a 2D array arr[-4 to +4][-4 to +4] stores a lower triangular matrix. Find the address of the location arr[-2][-1] if the starting address of the array is 500 and size of each element is 8 bytes. Assume, that elements are stored in column major order. ________________

Answer Key

1. (5)
2. (120)
3. (2292)
4. (1132)
5. (13)
6. (25)
7. (564)
8. (644)

Hints and Solutions

1. (5)
A lower triangular matrix is always a square matrix.
So, the number of rows in the array = 5.

2. (120)
Number of rows=Number of columns=7+7+1=15.

The sum of all elements-

= 15 +14 + 13 +……+ 3 + 2 + 1

= 120

3. (2292)
The address of the element arr[8][7] is-

$$= 1000 + \left(\frac{(8+16)(8+16+1)}{2} + (7+16)\right) \times 4$$

= 2292

4. (1132)
Number of non-zero elements in the -8th row = 15
Number of non-zero elements in the -7th row = 14

The address of arr[-6][4]-

= 1000 + (15+14+4)*4
= 1132

5. (13)
The element present at arr[1][2] in lower triangular matrix:
= 1 + 2 + 3 + 4 + 1 +1 +1
=13.

6. (25)
Number of elements in each row/column=3+3+1=7
The element present at arr[1][2] in upper triangular matrix:
= 7 + 6 + 5 + 4 + 1 + 1 + 1
= 25

7. (564)
Number of elements in each row= 4+4+1=9

When stored in column-major order, upper triangular matrix becomes lower triangular.

The number of non-zero elements from arr[-4][0] to arr[-1][-2]
= 1+2+3+3=9

The address of the element arr[-1][-2] is-

= 500 + (9-1)*8

= 564

8. (644)
Number of elements in each row= 4+4+1=9

When stored in column-major order, lower triangular matrix becomes upper triangular.

The number of non-zero elements from arr[-4][0] to arr[-2][-1]
= 9+8+2=19

The address of the element arr[-2][-1] is-

= 500 + (19-1)*8

= 644

BRANCH:CSE/IT Batch-Hinglish

Data structure &Programming
Arrays

DPP-02

[NAT]

1. Consider a lower triangular 2D array arr[][5] with 15 elements. The number of rows in arr is- ________

[NAT]

2.

Consider an integer 2D array a[-7 to +7] [-7 to +7] that stores an upper triangular matrix uppertm where uppertm[i][j] is 1 for all i>=j. The sum of all the elements in the array is ____________.

[NAT]

3. Consider an integer lower triangular 2D array arr[-16 to +15][-16 to +15] having base address 1000. If the size of the integer is 4 bytes, the address of the element arr[8][7]is- ____________

[NAT]

4. Consider an integer upper triangular 2D array arr[-8 to +7][-8 to +7] having base address 1000. If the size of integer is 4 bytes, the address of the element present at location arr[-6][4] is- ____________.

[NAT]

5. Consider the natural numbers starting from 1 are stored in a lower triangular matrix arr[-3 to 3][-3 to 3]. Find the element present at location arr[1][2]. _______________

[NAT]

6. Consider the natural numbers starting from 1 are stored in a upper triangular 2D array arr[-3 to 3][-3 to 3]. Find the element present at location arr[1][2]. ________________.

[NAT]

7. Consider a 2D array arr[-4 to +4][-4 to 4] stores an upper triangular matrix. Find the address of the location arr[-1][-2] if the starting address of the array is 500 and size of each element is 8 bytes. Assume that elements are stored in column-major order. _______________

[NAT]

8. Consider a 2D array arr[-4 to +4][-4 to +4] stores a lower triangular matrix. Find the address of the location arr[-2][-1] if the starting address of the array is 500 and size of each element is 8 bytes. Assume, that elements are stored in column major order. _______________

Answer Key

1. (5)
2. (120)
3. (2292)
4. (1132)
5. (13)
6. (25)
7. (564)
8. (644)

Hints and Solutions

1. (5)
A lower triangular matrix is always a square matrix.
So, the number of rows in the array = 5.

2. (120)
Number of rows=Number of columns=7+7+1=15.

The sum of all elements-

= 15 +14 + 13 +……+ 3 + 2 + 1

= 120

3. (2292)
The address of the element arr[8][7] is-

$$= 1000 + \left(\frac{(8+16)(8+16+1)}{2} + (7+16)\right) \times 4$$

= 2292

4. (1132)
Number of non-zero elements in the -8th row = 15
Number of non-zero elements in the -7th row = 14

The address of arr[-6][4]-

= 1000 + (15+14+4)*4
= 1132

5. (13)
The element present at arr[1][2] in lower triangular matrix:
= 1 + 2 + 3 + 4 + 1 +1 +1
=13.

6. (25)
Number of elements in each row/column=3+3+1=7
The element present at arr[1][2] in upper triangular matrix:
= 7 + 6 + 5 + 4 + 1 + 1 + 1
= 25

7. (564)
Number of elements in each row= 4+4+1=9

When stored in column-major order, upper triangular matrix becomes lower triangular.

The number of non-zero elements from arr[-4][0] to arr[-1][-2]
= 1+2+3+3=9

The address of the element arr[-1][-2] is-

= 500 + (9-1)*8

= 564

8. (644)
Number of elements in each row= 4+4+1=9

When stored in column-major order, lower triangular matrix becomes upper triangular.

The number of non-zero elements from arr[-4][0] to arr[-2][-1]
= 9+8+2=19

The address of the element arr[-2][-1] is-

= 500 + (19-1)*8

= 644

Data Structure

Hashing

DPP-01

[NAT]

1. Consider a hash table H with 512 slots. If 128 keys areto be stored in H, the load factor of H is ________________.

[NAT]

2. Consider a hash function that distributes keys uniformly. The hash table size is 2024. After hashing of how many keys will the probability that any new key hashed collides with an existing one exceed 0.75? ___________

[MCQ]

3. Suppose we are given n keys, m hash table slots, and two simple uniform hash functions h_1 and h_2. Further suppose our hashing schemes uses h_1 for the even keys and h_2 for the odd keys. What is the expected number of keys in a slot?

(a) $\frac{n}{m}$ (b) $\frac{m}{n}$

(c) $\frac{2n}{m}$ (d) $\frac{m}{2n}$

[NAT]

4. A hash table contains 9 buckets and uses linear probing to resolve collisions. The key values are integers and the hash function used is key % 9. If the values 41, 157, 72, 76, 31 are inserted in the table, in what location would the last key be inserted? ________________.

[MCQ]

5. Which one of the following hash functions on integers will distribute keys most uniformly over 10 buckets numbered 0 to 9 for i ranging from 0 to 2024?

(a) $h(i) = (12 * i) \bmod 10$

(b) $h(i) = (11 * i^2) \bmod 10$

(c) $h(i) = i^3 \bmod 10$

(d) $h(i) = i^2 \bmod 10$

[NAT]

6. Consider a double hashing scheme in which the primary hash function is $h_1(k) = k \bmod 17$, and the secondary hash function is $h_2(k) = 1 + (k \bmod 13)$. Assume that the table size is 17. Then the addressreturned by probe 2 in the probe sequence (assume thatthe probe sequence begins at probe 0) for key value k = 127 is____________.

[MCQ]

7. Consider a hash table with 11 slots. The hash function is h(k) = k mod 11. The collisions are resolved by chaining. The following 11 keys are inserted in the order: 28, 19, 15, 20, 33, 30, 42, 63, 60, 32, 43. The maximum, minimum, and average chain lengths in the hash table, respectively, are-

(a) 3, 0, 1 (b) 3, 3, 3

(c) 3, 0, 2 (d) 4, 0, 1

[MCQ]

8. A hash table of length 8 uses open addressing with hashfunction h(k)= 2 + k mod 8, and linear probing. After inserting 5 values into an empty hash table, the table isas shown below.

0	
1	
2	64
3	41
4	57
5	72
6	
7	29

How many different insertion sequences of the key values using the same hash function and linear probingwill result in the hash table shown above?

(a) 10 (b) 9

(c) 15 (d) 8

Answer Key

1. (0.25)
2. (1518)
3. (a)
4. (7)
5. (c)
6. (13)
7. (a)
8. (c)

Hints and Solutions

1. (0.25)

Load factor = $\frac{128}{512}$

2. (1518)

Let x be the number of keys after which any new key hashed collides with an existing one exceed $\frac{3}{4}$

$\frac{1}{2024} \times x = \frac{3}{4}$

$x = 1518$

3. (a)

If h_1 is used for the even keys and h_2 for the odd keys, the expected number of keys in a slot is still equivalent to $\frac{n}{m}$

4. (7)

41 mod 9 =5

157 mod 9 = 4

72 mod 9 = 0

76 mod 9 = 4

Since 157 occupies 4 and 41 occupies 5, 76 is hashed to 6

31 mod 9 = 4

Since 157 occupies 4, 41 occupies 5, 76 occupies 6, 31 is hashed to 7

5. (c)

$h(i) = i^3$ mod 10 will distribute keys most uniformly over 10 buckets numbered 0 to 9 for i ranging from 0 to 2024.

6. (13)

D-Hashing (k, probe) = (h_1(k) + probe × h_2(k)) mod m

Where, probe = 0 to m – 1

m = Number of empty slots in hash table)

D-Hashing (127, 2) = (127 mod 17 + 2 × (1 + 127 mod 13)) mod 17

= (8 + 2 × (1 + 10)) mod 17 = 13

7. (a)

28 mod 11 = 6

19 mod 11 = 8

15 mod 11 = 4

20 mod 11 = 9

33 mod 11 = 0

30 mod 11 = 8

42 mod 11 = 9

63 mod 11 = 8

60 mod 11 = 5

32 mod 11 = 10

43 mod 11 = 10

Maximum chain length = 3

Minimum chain length = 0

Average chain length = 11/11 = 1

8. (c)

64 is hashed to 2.

41 is hashed to 3.

57 is too hashed to 3. To avoid collision, 57 is hashed in 4.

Similarly, 72 and 29 are also dependent.

Order:

(i) 64 41 $\downarrow$ 57 single place

(ii) 41 64 $\downarrow$ 57 2 place

$\therefore$ After 57: (1) 64 41 57

(2) 41 57 64

(3) 41 34 57

(iii) 72.

(i) 64 41 57 $\downarrow$ 72 1 way $\Downarrow$ After 72

(ii) 41 57 64 $\downarrow$ 1 way

(iii) 41 64 57 $\downarrow$ 1 way

(i) $\underbrace{64\ 41\ 57\ 72}_{\text{After 29}}$ $\Downarrow$ 5ways

(ii) $\underbrace{41\ 57\ 64\ 72}_{\text{5 ways}}$

(iii) $\underbrace{41\ 64\ 57\ 72}_{\text{5 ways}}$

$\therefore$ Total = 15 ways

Data structure & Programming

Linked List

DPP-01

[NAT]

1. Consider a single linked list q with 2023 elements is passed to the following function:

```
struct node {
   int data;
   struct node *next;
};
void f(struct node *q){
    struct node *p;
    p=q->next;
   q->next=p->next->next;
}
```

The size of the linked list q after the execution of the function is _____________.

[MCQ]

2. Consider a single linked list q[‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’] is passed to the following function:

```
struct node {
   int data;
   struct node *next;
};
void f(struct node *q)
{
    struct node *p;
    p=q->next->next->next;
    q->next->next->next=p->next->next;
    p->next->next=q->next;
    printf("%c", p->next->next->next->data);
}
```

The output is-

(a) C (b) D

(c) E (d) B

[NAT]

3. Consider the following statements:

P: Linked Lists supports linear accessing of elements

Q: Linked Lists supports random accessing of elements.

Which of the following statements is/are INCORRECT?

(a) P only (b) Q only

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

[MCQ]

4. Consider a single linked list q[‘A’, ‘B’, ‘C’, ‘D’] is passed to the following function:

```
void f(struct node *q)
{
    if(q==NULL) return;
    f(q->next);
    printf("%c ", q->data);
}
```

The output is-

(a) C D B A (b) D C B A

(c) A B C D (d) B C D A

[NAT]

5. Consider the following statements:

P: Insertion at the end of the linked list is difficult than insertion at the beginning of the linked list.

Q: Deletion at the beginning of linked list is easier as compared to deletion at the end of the linked list.

Which of the following statements is/are CORRECT?

(a) Both P and Q (b) P only

(c) Q only (d) Neither P nor Q

[NAT]

6. The following C function takes a single-linked list p of integers as a parameter. It deletes the last element of the single linked list. Fill in the blank space in the code:

```
struct node {
   int data;
   struct node *next;
};
void delete_last(struct node *head)
{
   struct node *p=head, *q;
   if(!head) return;
   if(head->next==NULL){free(head);head=NULL;
   return;}
   while(______a_________){
      q = p;
      p=p->next;
   }
  ______b_________;
free(p);
 q=NULL; p=NULL;
}
```

(a) a: !head ; b: q->next = NULL;
(b) a: p->next ! = head ; b: q->next = q
(c) a: p->next ! = NULL ; b: q->next = NULL
(d) a: head->next ! = p ; b: q->next = p

[MCQ]

7. Consider a single linked list q[[‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’] is passed to the following function:

```
void func(struct node *head){
   struct node *p=head, *q=head;

   while(q!=NULL && q->next!=NULL && q->next->next != NULL){
      p=p->next;
     q=q->next->next;
   }
   printf(“%c”, p->data);
}
```

The output is-

(a) C (b) D
(c) E (d) B

[NAT]

8. The following C function takes a single-linked list p of integers as a parameter. It inserts the element at the end of the single linked list. Fill in the blank space in the code:

```
struct node
{
   int data;
   struct node *next;
};
void insert_last(struct node *head, struct node *q){
struct node *p=head;
if(!head) return;
while(______a______){
    p=p->next;
     ______b______;
    q=NULL;
    p=NULL;
   }
}
```

Assume, q is the address of the new node to be added.

(a) a: !head ; b: q->next = NULL;
(b) a: q->next ! = NULL; b: p->next = q
(c) a: p->next ! = NULL ; b: p->next = q
(d) a: head->next ! = p ; b: q->next = p

Answer Key

1. (2021)
2. (a)
3. (b)
4. (b)
5. (a)
6. (c)
7. (b)
8. (c)

Hints and Solutions

1. (2021)

The above function implementation skip the second and third elements. It connects the head element to the fourth element.

So, the size of the linked list is 2021.

2. (a)

A | 1	B | 2	C | ~~3~~ 5	D | 4	E | ~~5~~ 1	F | X
0	1	2	3	4	5

X represents NULL.
Initially, q points to node 0.

```
p=q->next->next->next;//p=3
q->next->next->next=p->next->next;//2->next=5
p->next->next=q->next;//4->next=1
printf("%c", p->next->next->next->data);
3->next->next->next->data
=4->next->next->data
=1->next->data
=2->data
=C
```

3. (b)

Linked List supports only linear accessing of elements.

4. (b)

```
void f(struct node *q){
if(q==NULL) return;
f(q->next);
printf("%c", q->data);
}
```

A | 2	B | 3	C | 4	D | X
1	2	3	4

X represents NULL.
f(1):
1 is NOT NULL.
f(2).
P4: It prints 1->data i.e A.
f(2):
2 is NOT NULL.
f(3).
P3: It prints 2->data i.e B.
f(3):
3 is NOT NULL
f(X).
P2: It prints 3->data i.e C.
f(4):
4 is NOT NULL;
f(X).
P1: It prints 4->data i.e D.
f(X):
X is equal to NULL. So it returns to f(4);
OUTPUT: D C B A

5. (a)

P: CORRECT. Insertion at the end of the linked list is difficult than insertion at the beginning of the linked list.

Q: CORRECT. Deletion at the beginning of linked list is easier as compared to deletion at the end of the linked list.

6. (c)

```
void delete_last(struct node *head)
{
   struct node *p=head, *q;
  if(!head) return;
  if(head->next==NULL){free(head);head=NULL;
   return;
}
while(p->next!=NULL)
{
    q = p;
    p=p->next;
 }
 q->next=NULL;
 free(p);
 q=NULL; p=NULL;
}
```

7. (b)

The code prints the middle element in the linked list q.

A B C D E F G.

Output: D

8. (c)

```
void insert_last(struct node *head, struct node *q){
struct node *p=head;
if(!head) return;
while(p->next!=NULL)
p=p->next;
p->next=q;
q=NULL;
p=NULL;
}
```

Data structure & Programming
Linked List

DPP-02

[NAT]

1. Consider a linked list [a, b, c, d, e]. ptr is a pointer pointing to the head/start node. Assume a node in the linked list is defined as:

```
struct node
{
int data;
struct node *next;
};
```

The output of the statement printf("%d", ptr->next->next->data) is ____.

[NAT]

2. Consider a single linked list of integers [9, 8, 7, 6, 5, 4,3] is passed to the following function:

```
struct node
{
int data;
struct node *next;
};
int  func(struct node *q){
static int k=0;
struct node *ptr=q;
if(!ptr) return 0;
 else if(ptr->next==NULL) return k+=ptr->data;
 else{
 k+=ptr->data;
  func(ptr->next);
 return k;
 }
 }
```

Assume, q points to head/start node in the linked list. The value returned by func(q) is __________.

[MCQ]

3. Consider the following function:

```
struct node
{
int data;
struct node *next;
};
struct node * f(struct node *head, int k){
struct node *p=head;
int i=0;
while(i<k/2){
p=p->next;
i++;
 }
return p;
}
```

Assume head points to the start node of the linked list and k is the number of elements in the linked list, the function returns-

(a) The pointer to the middle element in the linked list.
(b) The pointer to the second element in the linked list.
(c) The middle element in the linked list.
(d) The second last element in the linked list.

[MCQ]

4. Consider the following function:

```
struct node
{
int data;
struct node *next;
};
void f(struct node *head){
struct node *a=head, *b=NULL, *c=NULL ;
while(a){
c=a->next;
a->next=b;
b=a;
a=c;
 }
head=b;
}
```

Assume head points to the start node of the linked list, the function-

(a) Sorts the linked list.
(b) Interchanges every two consecutive elements in the list.

(c) Reverses the list.
(d) None of the above.

[MCQ]

5. Consider a single linked list [1, 2, 3, 4, 5] is passed to the following function:

```
struct node
{
int data;
struct node *next;
};
void  func(struct node *p){
struct node *q=p->next, *temp;
if(!p||!(p->next)) return;
else{
temp=q->data;
q->data=p->data;
p->data=temp;
func(p->next->next);
}
}
```

Initially, the address of the head/start node is passed to the function func(*p), the arrangement of the linked list after function execution is –

(a) 2 3 4 5 1 (b) 5 4 3 2 1
(c) 2 1 4 3 5 (d) 2 1 4 5 3

[MSQ]

6. Consider the following function:

```
struct node
{
int data;
struct node *next;
};
struct node * f(struct node *head){
struct node *p=head, *q=head;
while(q!=NULL && q->next!=NULL && q-
>next->next != NULL){
p=p->next;
q=q->next->next;
if(p==q) break;
}
return p;}
```

Assume head points to the start node of the linked list, the function-

(a) Returns the pointer to the node where a cycle/loop ends (assume the leftmost node in a loop is the start node of a cycle).
(b) Returns the pointer to the node where a cycle/loop starts (assume the leftmost node in a loop is the start of a cycle).
(c) Reverses the list.
(d) Detects a cycle in the list.

[MCQ]

7. Consider the following function:

```
struct node
{
int data;
struct node *next;
};
void f(struct node *head, int e){
struct node *p, *q;
if(head->data==e){
q=p;
p=p->next;
_________;
head=p;
return;
}
q=head; p=head->next;
while(p->next!=NULL){
if(p->data==e){
______________;
free(p);
return;
}
q=p;
p=p->next;
}
if(p->data==e){
q->next=NULL;
free(p);
}
}
```

Assume there are at least two elements in the single linked list of integers. The starting node's address is contained in the head pointer passed to the function. The function f() searches for the element e in the list. If found, the function deletes the node. The missing statements are-

(a) free(q), q->next=p->next
(b) free(q), q=p->next
(c) free(p), q=p->next
(d) free(p), q->next=p->next

[MCQ]

8. A node of a linked list is to be created by calling malloc() function. The malloc() returns NULL if-

(a) Stack overflow occurs.

(b) Memory leakage occurs.

(c) Memory is full.

(d) None of the above.

Answer Key

1. (99)
2. (42)
3. (a)
4. (c)
5. (c)
6. (b, d)
7. (a)
8. (c)

Hints and Solutions

1. **(99)**
printf("%d", ptr->next->next->data); //The ASCII of c is printed i.e 99

2. **(42)**
It computes the sum of all the data elements in a recursive manner.
Output = 9+8+7+6+5+4+3=42

3. **(a)**
It returns the pointer to the middle element in the linked list.

4. **(c)**
It reverses the list.

5. **(c)**
The function reverses the elements in groups of 2.
So, output is 2 1 4 3 5

6. **(b, d)**
It detects a cycle in the linked list and returns the pointer to the node where the loop starts from.

7. **(a)**

```
void f(struct node *head, int e){
struct node *p, *q;
if(head->data==e){
q=p;
p=p->next;
free(q);
head=p;
return;
}
q=head; p=head->next;
while(p->next!=NULL){
if(p->data==e){
q->next=p->next;
free(p);
return;
}
q=p;
p=p->next;
}
if(p->data==e){
q->next=NULL;
free(p);
}
}
```

8. **(c)**
The malloc() returns NULL only if the memory is full.

Data Structure & Programming

Stacks and Queues

DPP-01

[NAT]

1. Consider the following sequence of operations on an empty stack:

push (5); push (2); pop(); push(4); push(6); p=pop(); q=pop(); r=pop();

The value of p+q-r is-________.

[MCQ]

2. Which of the following includes the applications of stack?

(a) Recursive function calls
(b) HTML and XML Tag matching
(c) Checking if an expression contains balanced parantheses.
(d) Finding the maximum element in a given sequence.

[NAT]

3. A stack is implemented using array. S represents the pointer to the top element in the stack. Initially the stack contains the elements: a(top), b. Assume Push(S, i) push an element i into the stack at index S. Whenever a Push operation will be performed, it will returns S++ after the push operation. Pop() pops the topmost element and returns the next top index. Top() is a function that returns the topmost element of the stack. Consider the following statements:

P: Top(Pop((Pop(Pop((Push(Push(S, c), d)))))))= a

Q: Pop(Pop(Pop(Pop(Push(Pop(Push(S, c)), d))))) = a

Which of the following statements is/are INVALID?

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

[MCQ]

4. A single array A[1…MAXSIZE] is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables top1 and top2 (topl< top 2) point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for "stack full" is-

(a) (top1=MAXSIZE/2)and(top2=MAXSIZE/2+1)
(b) (top1=MAXSIZE/2)or(top2=MAXSIZE/2+1)
(c) top1 + top2 = MAXSIZE
(d) top1=top2-1

[NAT]

5. A stack is implemented using a singly linked list that uses node structure-

```
struct node
{
 int data;
struct node *next;
}
node;
```

Let *head denote the address of the start node respectively. Assume, the stack is not empty. Consider the following function that intends to delete the topmost element of the stack:

```
node * f(node *head)
{
     node *p=head; ____________;
     free(p);
     p=NULL;
  }
```

The missing blank is-

(a) while(p → next!=NULL) p=p→next;
(b) p=p → next;
(c) head=head → next;
(d) None

[MSQ]

6. Which one of the following permutations cannot be obtained in the output string using a stack and assuming that the input sequence is a, b, c, d, e in the same order?

(a) c d e a b (b) a e b c d
(c) c d e b a (d) e d c b a

[MCQ]

7. A stack is implemented using array of size 4. S represents the pointer to the top element in the stack. Initially the stack contains the elements- a(top), b. Assume Push(S, i) push an element i into the stack at index S. Whenever a Push operation will be performed, it will returns S++ after the push operation. Pop() pops the topmost element and returns the next top index. isEmpty() returns TRUE if the stack is empty. isFull() returns TRUE if the stack is full. Consider the following statements:

P: isFull(Push(Pop(Push(Push(S, c), d))), e))= TRUE

Q: isEmpty(Push(Pop(Pop(Push(Pop(Push(S, c)), d))))), e) = FALSE

Which of the following statements is/are VALID?

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

[NAT]

8. Let S be a stack of size n≥1. Starting with the empty stack, suppose we push the first *5* natural numbers in sequence, and then perform 5 pop operations. Assume that Push and Pop operations take 3 seconds each, and 1 seconds elapse between the end of one such stack operation and the start of the next operation. The average stack-life of an element of this stack is __________.

Answer Key

1. (5)
2. (a, b, c)
3. (c)
4. (d)
5. (c)
6. (a, b)
7. (c)
8. (17)

Hints and Solutions

1. (5)

push (5);
push (2);
pop(); //2 is popped
push(4);
push(6);
p=pop(); //6 is popped
q=pop(); //4 is popped
r=pop(); //5 is popped

The final value of p+q-r = 6+4-5= 5

2. (a, b, c)

The application of stack:
Recursive Function Calls
HTML and XML Tag matching
Checking if an expression contains balanced parantheses.

3. (c)

Stack already contains a(top), b.
top(pop((pop(pop((push(push(S, c), d)))))))
It pushes c into the stack. It pushes d into the stack.
It pops d. It pops c.
It pops a.
Top contains b now.
pop(pop(pop(pop(push(pop(push(S, c)), d))))) = a
It pushes c and pops it.
It pushes d and pops it.
It pops a and then b.
Last pop operation cannot be implemented as the stack is already empty.

4. (d)

If the stacks are growing from two ends, top2-top1=1.

5. (c)

If a stack is implemented using linked list, push and pop occurs from one default location-head/start of the linked list.

The missing statement include-
head=head->next

6. (a, b)

(a) is NOT possible
Push a;
Push b;
Push c;
Pop c;
Push d;
Pop d;
Push e;
Pop e;
Pop a; (Not possible) top contains b.

(b) is NOT possible.
Push a;
Pop a;
Push b;
Push c;
Push d;
Push e;
Pop e;
Pop b;(Not possible) top contains d.

7. (c)

isFull(push(pop(push(push(S, c), d))), e))= TRUE

Push c;
Push d;
Pop;
Push e;
The stack is full (b, a, c, e(top)).
is Empty(push(pop(pop(push(pop(push(S, c)), d))))), e) = FALSE
Push c;
Pop c;
Push d;
Pop d;
Pop a;
Push e;(Stack contains b, e(top))
The stack is not empty.

8. **(17)**

Stack Life time of 5=1
Stack Life time of =2*3+3*1=9
Stack Life time of 3=4*3+5*1=17
Stack Life time of 2=6*3+7*1=25
Stack Life time of 1=8*3+9*1=33
Average stack-life of an element =
(1+9+17+25+33)/5 = 17

Data Structure

STACKS AND QUEUES

DPP-02

[MCQ]

1. Consider the following infix expression:
P-Q/(R*S)+T*U
The prefix notation of the given expression is-
(a) -+P/Q*RS*TU (b) +-P/Q*RS*TU
(c) +-/PQ*RS*TU (d) None of the above.

[MCQ]

2. Consider the following expression:
P+Q/R-S*T^U/V-W
The post fix notation of the given expression is-
(a) PQR/+STU^*V/-W-
(b) PQ+RS-TU^*V/-W-
(c) PQR/-STU^*V/W+-
(d) None of the above

[MCQ]

3. Consider the following prefix notation:
/^*+abc/de^gh
The postfix notation of the given expression is-
(a) ab+c*de/^gh^/
(b) abc+*de/^gh^/
(c) abc+de/*^gh^/
(d) None of the above

[NAT]

4. Consider the following infix expression:
P+Q/R-S*T^U/V-W
The maximum size of the operator stack required to convert the given infix to postfix notation is
_________.

[MCQ]

5. Consider the following infix expression:
P*Q/R-S*T+U/V*W
On reaching the symbol V, the top two contents of the operator stack are:
(a) /, * (b) /, -
(c) *, + (d) /, +

[NAT]

6. Consider the following postfix expression:
8 2 3 ^/ 5 3 *+ 2 1 / -
The result of evaluating the above postfix expression is _________.

[NAT]

7. Let X be the result when the below postfix expression is evaluated:
X = 8 3 1 + - 2 ^ 7 1 2 - * +
And Y be the result of the following postfix expression:
Y= X 3 / 4 +
The value of $(X+Y)^{0.5}$ is _________

[NAT]

8. Let X be the result when the below postfix expression is evaluated:
X = 4 5 1 + * 2 / 3 1 2 + * +
Let Y be the maximum size of the operand stack, the value of X-Y is ________

Answer Key

1. (b)
2. (a)
3. (a)
4. (3)
5. (d)
6. (14)
7. (4)
8. (17)

Hints and Solutions

1. (b)

P-Q/(R*S)+T*U
P-Q/*RS+T*U
P-/Q*RS+*TU
-P/Q*RS+*TU
+-P/Q*RS*TU

2. (a)

P+Q/R-S*T^U/V-W
P+Q/R-S*TU^/V-W
P+QR/-S*TU^/V-W
P+QR/-STU^*/V-W
P+QR/-STU^*V/-W
PQR/+-STU^*V/-W
PQR/+STU^*V/--W
PQR/+STU^*V/-W-

3. (a)

/^*+abc/de^gh
/^*+abc/degh^
/^*+abcde/gh^
/^*ab+ c de/ gh^
/^ab+c* de/ gh^
/ ab+c*de/^ gh^
ab+c*de/^gh^/

4. (3)

^
/ * /
+ - -

Postfix notation: PQR/+STU^*V/-W-

5. (d)

* /
* / - +

Post fix notation till symbol V is encountered: PQ*R/ST*-UV

The top two contents of the stack are - /, +

6. (14)

3	3		1		
2	8	5	15	2	2
8	1		16	14	

PUSH(8);
PUSH(2);
PUSH(3);
^ is encountered. Pop(3), Pop(2)
PUSH(2^3) i.e PUSH(8)
/ is encountered. Pop(8), Pop(8)
PUSH(8/8) i.e PUSH(1);
PUSH(5);
PUSH(3);
* is encountered. Pop(3), Pop(5)
PUSH(5*3) i.e PUSH(15)
+ is encountered. Pop(15), Pop(1)
PUSH(1+15) i.e PUSH(16)
PUSH(2);
PUSH(1);
/ is encountered. Pop(1), Pop(2)
PUSH(2/1) i.e PUSH(2);
- is encountered. Pop(2), Pop(16)
PUSH(16-2) i.e PUSH(14);
Final Result is 14. It is available in the stack.

7. (4)

8 3 1 + - 2 ^ 7 1 2 - * +
8 4 - 2 ^ 7 1 2 - * +
4 2 ^ 7 1 2 - * +
16 7 1 2 - * +
16 7 -1*+
16 -7 +
X = 9
Y = 9 3 / 4 +
Y = 3 4 +
Y = 7
$(X+Y)^{0.5} = 16^{0.5} = 4$

8. **(17)**

			2		
1			1	3	
5	6	2	3		9
4		24	12		21

X= 21, Y=4, X-Y=17

Branch : CSE/IT **Batch : Hinglish**

Data Structure & Programming

Tree

DPP-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	$\frac{n+1}{2}$

Branch : CSE/IT **Batch : Hinglish**

Data Structure
Tree

DPP-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

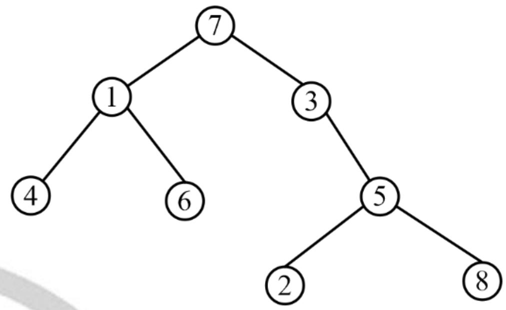
[MCQ]

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-

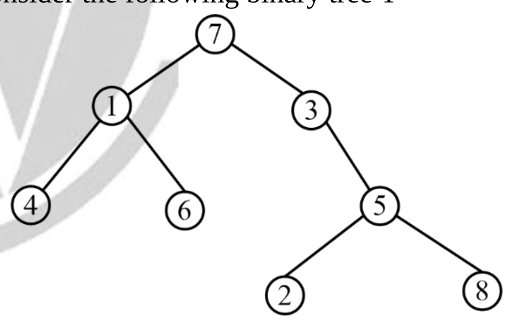


The in-order traversal of T is-

(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-

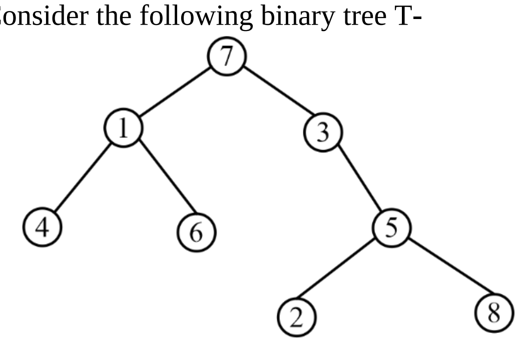


The pre-order traversal of T is-

(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-

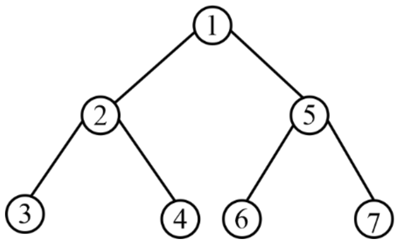
(a) 1, 2, 4, 7, 9, 8, 5, 3, 6
(b) 1, 2, 4, 7, 8, 9, 5, 3, 6
(c) 1, 2, 3, 4, 5, 6, 7, 8, 9
(d) None of the above.

Answer Key

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

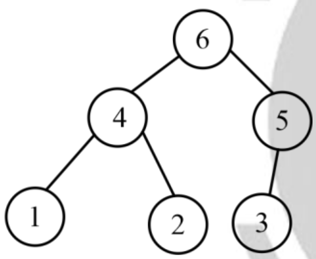
Hints and Solutions

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)**



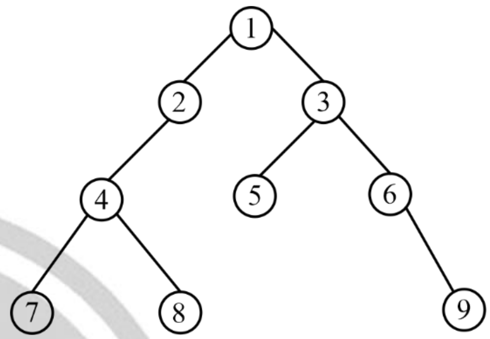
In order: 1 4 2 6 3 5
Pre-order: 6 4 1 2 5 3
Post-order: 1 2 4 3 5 6
Clearly, LASTIN ≠ LASTPRE ≠ LASTPOST

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

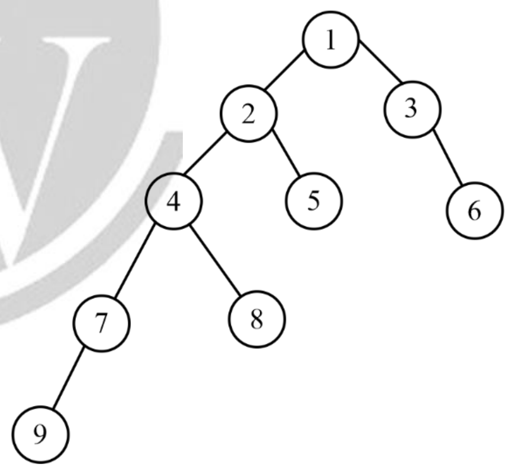
6. **(c)**
The post-order traversal of T is - 4 6 1 2 8 5 3 7

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 : 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)

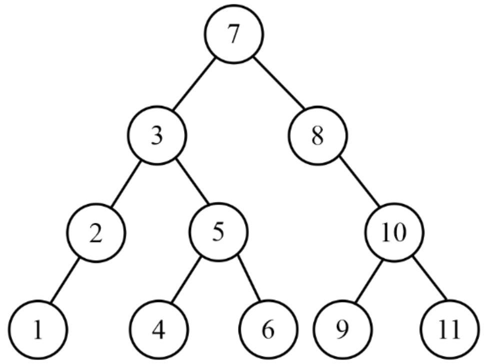
Pre-order traversal of BST:

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);
}
```

Data Structure
Tree

DPP-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:

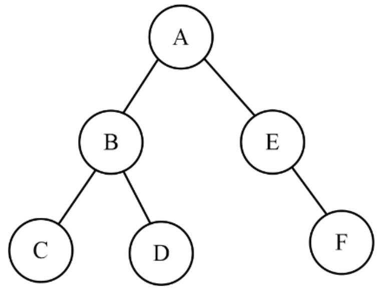
```
struct treenode
{
    struct treenode *left;
    int data;
    struct treenode *right;
 };
int func(struct treenode *t)
{
      if(t==NULL) return 0;
      elseif(t->left==NULL && t->right==NULL)
      return 1;
      else
      return 1+func(t->left)+func(t->right);
}
```

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 : Hinglish**

Data Structure

Tree

DPP 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

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

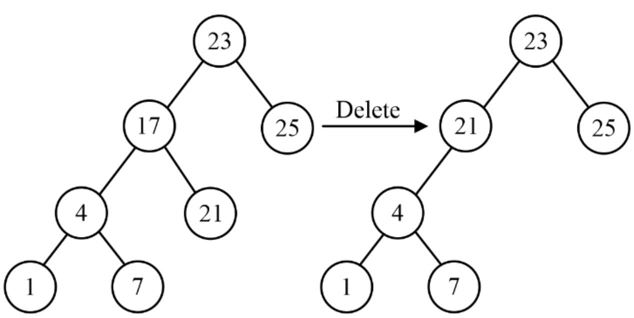
Now, T(3) = 2

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

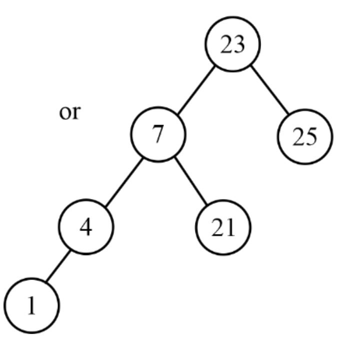
4. (a)

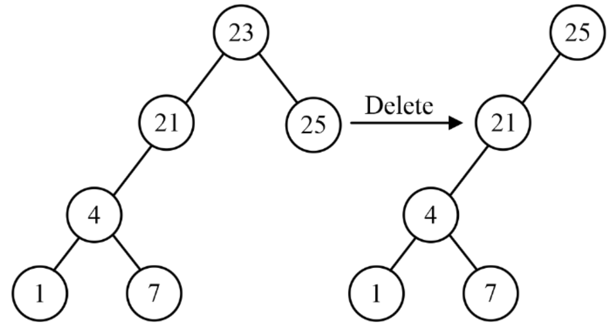
If the root node of a BST is deleted, it can be replaced by inorder predecessor/successor.

5. (a, b, c)

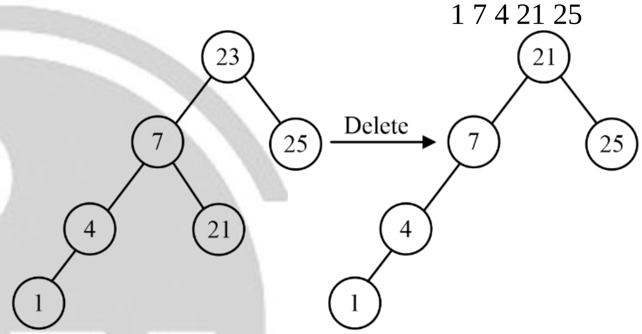


or



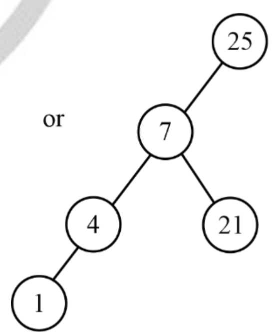


Post order traversal: -
1 7 4 21 25



Post order traversal: -
1 4 7 25 21

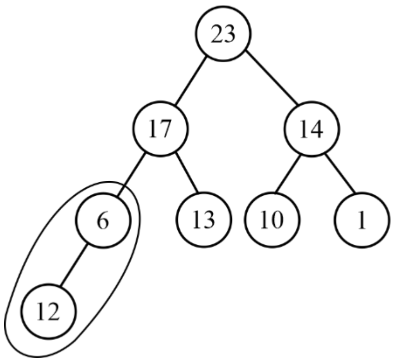
or



Post order traversal: -
1 4 21 7 25

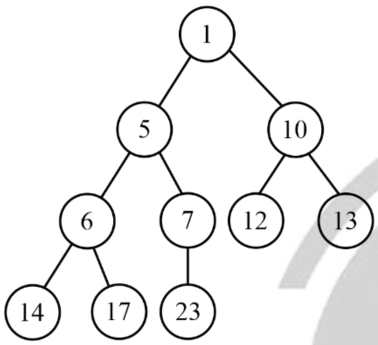
6. (b, c)

(a)



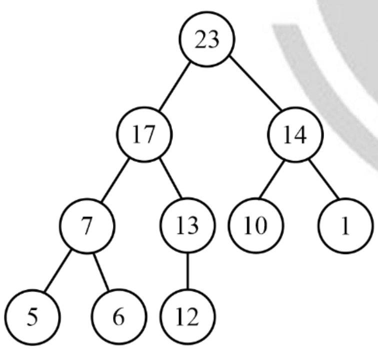
Not possible defies max-heap property

(b)



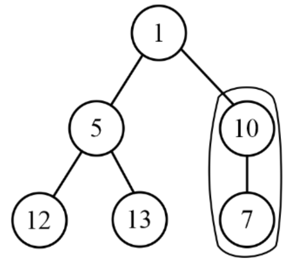
Satisfies min-heap property

(c)



Satisfies max-heap property.

(d)



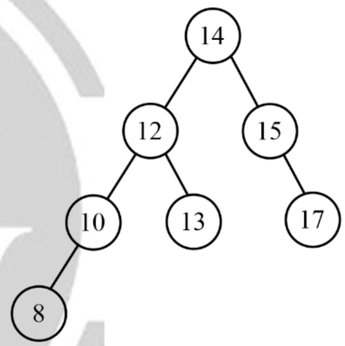
Not possible defies min-heap property

7. (0)

Both the statements are CORRECT.

8. (12)

Resultant AVL tree:



Branch : CSE/IT Batch : Hinglish

Data Structure

Tree

DPP-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

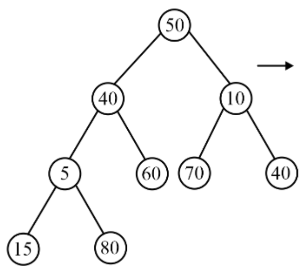
Maximum number of comparisons to find the maximum element in a min heap of 1024 elements = 512 – 1 = 511.

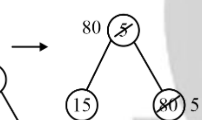
2. (10)

0	1	2	3	4	5	6	7	8
50	40	10	5	60	70	40	15	80

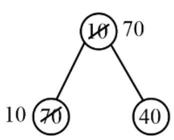
Internal nodes → Needs to be adjusted

leaf nodes → Already in max-heap

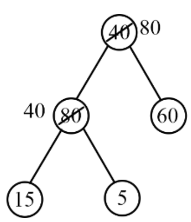




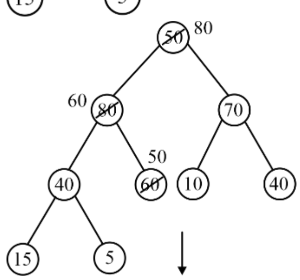
Number of comparisons
{(5, 15), (15, 80)}
⇓
2 comparison 1 swap



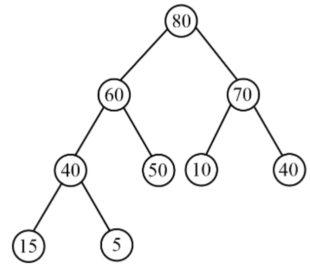
Number of comparisons
{(10, 70), (70, 40)}
⇓
2 comparison 1 swap



Number of comparisons
{(40, 80), (80, 60)}
⇓
2 comparison 1 swap



Number of comparisons
{(50, 80), (80, 70)}
{(50, 40), (50, 60)}
⇓
4 comparison 2 swap

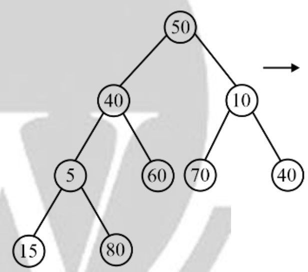


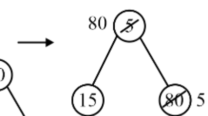
3. (5)

0	1	2	3	4	5	6	7	8
50	40	10	5	60	70	40	15	80

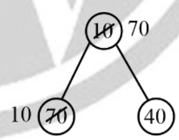
Internal nodes → Needs to be adjusted

leaf nodes → Already in max-heap

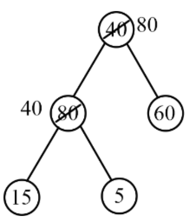




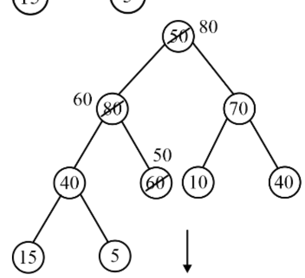
Number of comparisons
{(5, 15), (15, 80)}
⇓
2 comparison 1 swap



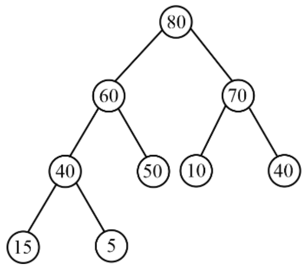
Number of comparisons
{(10, 70), (70, 40)}
⇓
2 comparison 1 swap



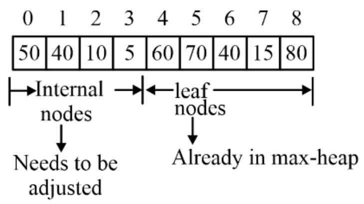
Number of comparisons
{(40, 80), (80, 60)}
⇓
2 comparison 1 swap

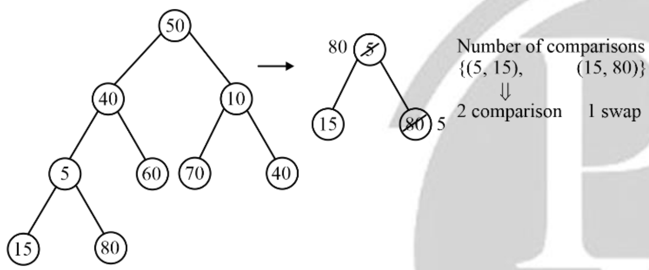


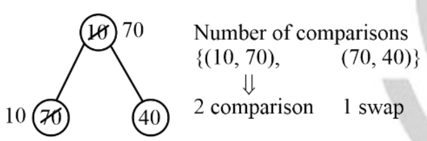
Number of comparisons
{(50, 80), (80, 70)}
{(50, 40), (50, 60)}
⇓
4 comparison 2 swap

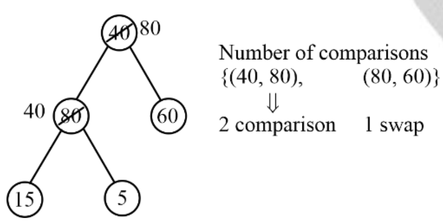


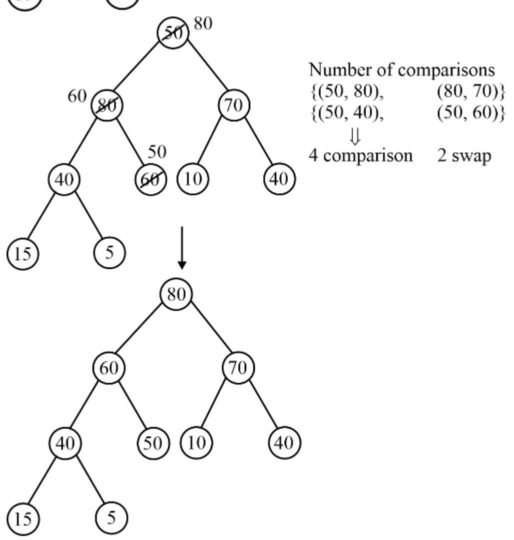
4. (a)



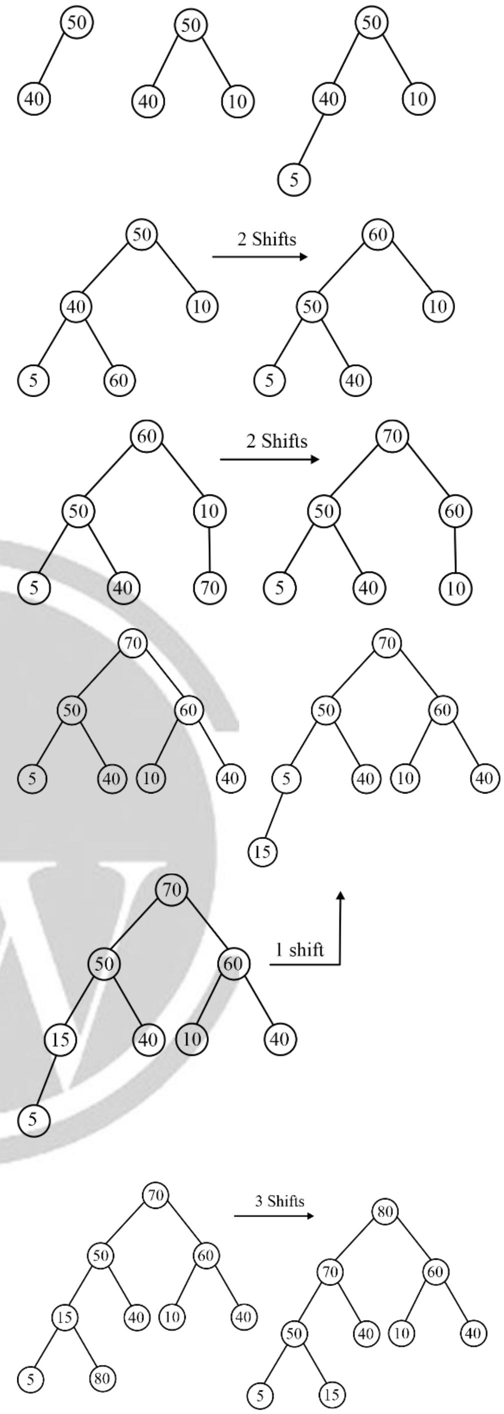








5. (8)



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 : Hinglish**

Data Structure

Tree

DPP-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.

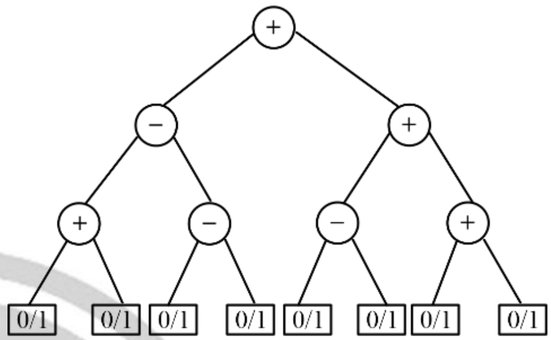
S_1: The sequence of procedure calls corresponds to a preorder traversal of the activation tree.
S_2: The sequence of procedure returns corresponds to a postorder traversal of the activation tree.

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)**

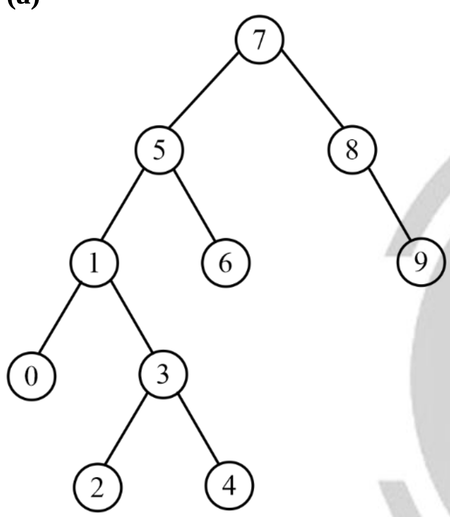
Hints and Solutions

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 $O(n^2 \log n)$

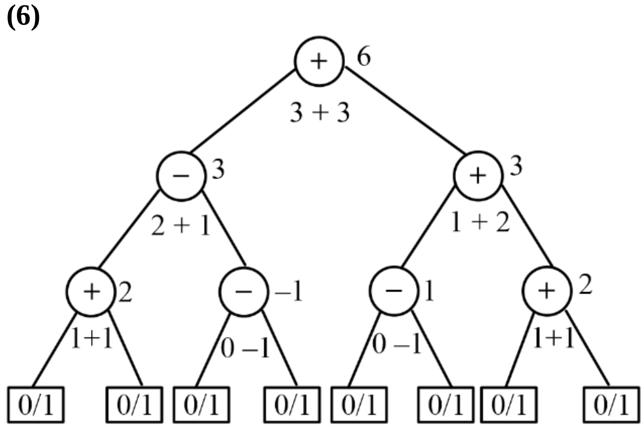
3. (a)



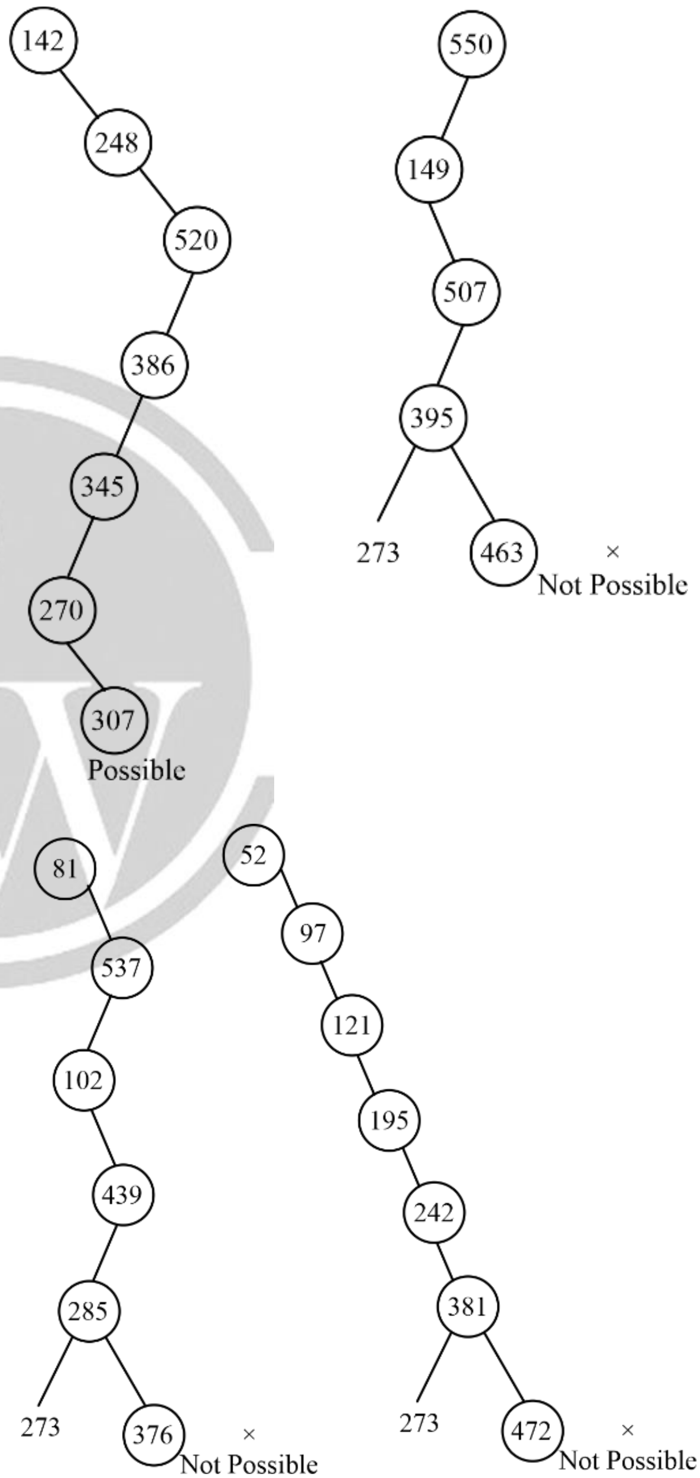
Preorder traversal
7 5 1 0 3 2 4 6 8 9

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

5. (6)



6. (d)



7. (5)

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

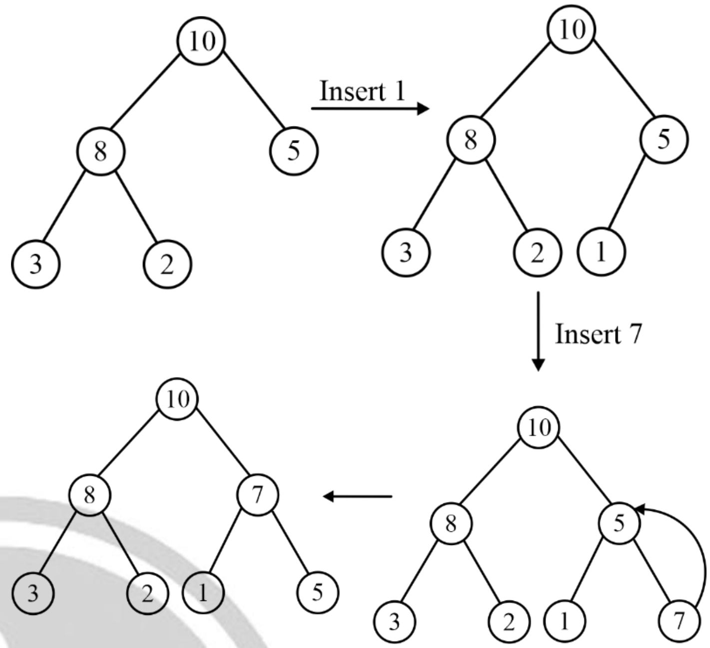
$L = 41,\ \ I = 10 -$

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

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

$N = 4 + 1 = 5$

8. (a)



Level order Traversal:

10 8 7 3 2 1 5