

[This question paper contains 8 printed pages.]

Your Roll No.....

Sr. No. of Question Paper : 1527

G

Unique Paper Code : 2342012301

Name of the Paper : Data Structures

Name of the Course : **B.Sc. (H) Computer Science**

Semester : III (DSC)

Duration : 3 Hours

Maximum Marks : 90

Instructions for Candidates

1. Write your Roll No. on the top immediately on receipt of this question paper.
2. **All** questions in **Section A** are compulsory.
3. Attempt any **four** questions out of **six** in **Section B**.

Section – A

1. (a) What will be the maximum number of nodes in a Binary tree with depth d ? (2)

P.T.O.

- (b) What would be the output after the following code segment is executed on the following linked list

10 → 30 → 20 → 90 → 100

```
node *p;
```

```
p = head;
```

```
while (p->next->next)
```

```
{
```

```
    p=p-> next;
```

```
}
```

```
cout<< p-> data; (2)
```

- (c) For the given queue - a, b, c, d, List the sequence of enqueue and dequeue operation to obtain the following queue : d, c, b, a (2)

- (d) Store the following univariate polynomial in a singly linked list

$$5x^3 - 6x^2 - 3x + 2 \quad (3)$$

- (e) Solve the following recurrence relation using Master's theorem-

$$T(n) = 2T(n/2) + n \log n \quad (3)$$

- (f) Write a function for finding the minimum element in a Binary Search tree (3)

- (g) void fun(int x)

```
{
```

```
    if (x>0)
```

```
{
```

```
    fun( x-1);
```

```
    cout << x;;
```

```
    fun(x-1);
```

```

    }

    }

void main()

{

    fun(3);

}

```

What will be the output for the above code snippet? Create a recursion tree for the same.

(5)

- (h) Convert the following expression to postfix using a stack. Show the status of stack after each operation

$$(A+B) * (C-D) / F - X*Y / Z$$

(5)

- (i) Given a two dimensional array $A[5][10]$. A is stored at address is 1000 and each element is of size 4. Find the address of $A[3][5]$ using both row major and column major order.

(5)

Section - B

2. (a) What is the advantage of using a doubly linked list over a singly linked list? Write a function to remove duplicate elements from a sorted doubly linked list. (2+6)

- (b) Draw a binary tree T such that (5+2)

- Each node of T stores a single character
- A preorder traversal of T yields *mnopxyz*
- An inorder traversal of T yields *npomyzx*

Also give the post-order traversal of T

3. (a) Write a function to count all the occurrences of an element in an integer type array. (5)

- (b) Evaluate the following postfix expression using a stack. Show every step.

$$4\ 3\ *\ 2\ /\ 1\ 8\ 2\ \wedge\ \wedge\ +\ 2\ -\ 3\ +$$

(\wedge operator depicts exponent) (5)

P.T.O.

- (c) Draw the expression tree for the following : (5)

$$2 + 5 * 3^2 + 9 + ((1+8) * 3)$$

4. (a) For a given set of values, write a program for printing the Next Greater Element (NGE) for every element. Use stack data structure. Write the NGE for the following elements 4,5,2,25. Also, explain the time complexity of the code. (6+2+2)
- (b) Write a recursive function for computing nth Fibonacci number via binary recursion. (5)
5. (a) Create a class to implement ADT queue using two stacks. Write functions - enqueue, dequeue and empty. (3+5)
- (b) Let $f(n) = 3n^2 + 4n \log n + 5n$. Show that $f(n) \in O(n^2)$. (3)

- (c) Sort the following array using insertion sort

10, 9, 11, 7, 8, 5

Show the status of array after each iteration.

(4)

6. (a) Create an AVL tree by inserting the following items successively. Show the status of tree after each insertion

10, 15, 50, 12, 11, 30, 40, 60, 7

Delete 15 from the above tree. (6+4)

- (b) Write a function to return the i^{th} element of a singly linked list. Compare the time complexity of accessing an element in an array and a linked list. (3+2)

7. (a) Consider a max heap with following values-

50, 30, 20, 15, 10, 8, 16

(i) Insert a new node with value 60.

(ii) Delete a node with value 50. (5)

P.T.O.

- (b) Write a function to delete the element pointed by the cursor in a circular linked list. (5)
- (c) Write a recursive function to find the total number of nodes in a Binary Search Tree. (5)