(b) Insert the following elements one by one into an empty Max Heap and show the heap after each insertion :

25, 40, 15, 60, 10, 50 (5)

(c) Construct a Binary Search Tree using the following data elements.

45, 25, 65, 15, 35, 55, 75, 10, 20

Delete node 25 and show the final BST. (5)



[This question paper contains 8 printed pages.]

**Your Roll No.....**

**Sr. No. of Question Paper : 2347** **L**

Unique Paper Code : 2342571201

Name of the Paper : Data Structures

Name of the Course : **B.A. (P) With CS as Minor/ Non-Major, B.Sc (Phy. Sc.)/ SOL**

Semester : II

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. **Section A** is compulsory.
3. Attempt any **four** questions from **Section B**.
4. Parts of a question must be answered together.

P.T.O.

## Section A

1. (a) Define Big omega notation. (2)

(b) Mention any two applications of Queue. (2)

(c) Arrange the following Asymptotic Functions in increasing order: (3)

$n!$, $2^n$, $n^3$, $n \log n$, $\log^2 n$

(d) Convert the following infix expression into postfix expression. Show all the intermediate steps : (3)

(A – B / C) * (D + E)

(e) Depict the scenario when a recursive function has no base condition? (3)

(f) Differentiate between Binary Heap and Binary Search Tree in term of complexity. (3)

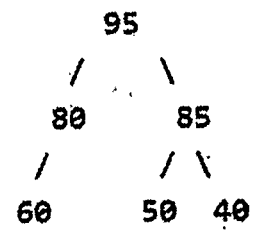
(g) Which pointer updates are required when inserting a node at the beginning of a Singly Linked List? Write the process step by step to insert a new node at the end of Singly Linked List. (4)

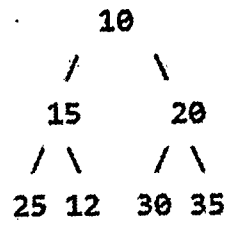


(b) (i) Write the steps to delete the Root node in a Max Heap.

(ii) Two Binary Trees are given below. Determine whether each tree is a Binary Heap or not.

Justify your answer with the help of suitable reason. (5)



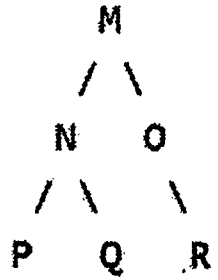

(c) Construct the binary tree from the following given Inorder and Postorder traversal. (5)

Inorder : 4 2 5 1 6 3 7

Postorder : 4 5 2 6 7 3 1

7. (a) Define height of a binary tree. Write a C++ function to compute it. (5)

(b) Express the following Binary Tree using Preorder, Inorder, and Postorder traversal : (5)




(c) A stack is implemented with a maximum size of 5. Perform the following operations: (5)

push(15), push(13), pop(), push(17), push(72), pop(), push(12), size(), peek()

Explain what happens at each step, the final contents of the stack, and how overflow or underflow conditions are handled.

6. (a) Perform the following operations on an Array of size 5. (5)

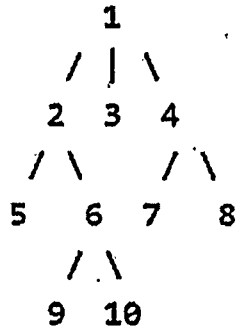
(i) Insert the elements: 40, 25, 30, 60, 15.

(ii) Delete an element 30 from the list.

(iii) Display the current list of elements after the insertions and deletions.



(h) Consider the following tree : (5)




Based on the above tree, answer the following :

(i) Identify the siblings of node 6.

(ii) List all ancestors of node 9.

(iii) List all descendants of node 2.

(iv) Find the degree of root node.

(v) Find the height of the tree.

(i) Compare the time complexity of searching an element in : (5)

(i) Array

(ii) Linked List

(iii) Binary Search Tree

(iv) Balanced Binary Search Tree

Which data structure is most efficient for searching and why?

## Section B

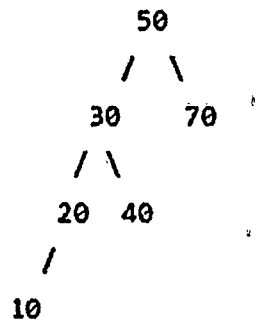
2. (a) Solve the following recurrence relation using Master's Theorem (5)

$T(n)= 8T (n/2) + n^2$.

(b) Explain the binary recursion with the help of a suitable example. (5)

(c) Explain the significance of balancing a Tree. Is the following binary tree an AVL Tree? (5)

If not, identify the node(s) violating AVL property:




3. (a) Compare Singly Linked List and Doubly Linked List with the help of suitable diagrams. (5)

(b) Write a recursive code to compute the factorial of a number. Trace the execution for n=5. (5)



(c) Evaluate the following prefix expression using stack. Show the stack status after each step:

* + 4 5 2 / 8 4 (5)

4. (a) Determine the number of nodes in a full binary tree of height 4. Also calculate the number of leaf nodes. (5)

(b) Show that $4n^3 + 3n^2 = 0(n^3)$ and find the value of c and $n_0$. (5)

(c) Find the output of the following recursive function for fun (4) after execution : (5)

```
int fun(int n)
{
    if(n==0)
        return 0;
    return n + fun(n–1);
}
```

5. (a) A circular queue of size 5 is initially empty. Perform the following operations and show Front/ Rear after each step : (5)

ENQUEUE(11), ENQUEUE(22), ENQUEUE(33), DEQUEUE(), ENQUEUE(44), ENQUEUE(55)