(Please write your Exam Roll No.) *Exam Roll No. … … … … … … … …*

Bharati Vidyapeeth's
Institute of Computer Applications and Management
A-4, Paschim Vihar, New Delhi-63

Model Question Paper - III [MCA – II Sem.]

Paper Code: MCA–102 **Subject: Data and File Structures**

Time: 3 Hours **Maximum Marks: 75**

Note: Attempt five questions in all. Question no. 1 is compulsory and attempt one question from each unit.

1. Answer the following questions, briefly:- (2.5×10 = 25)

(a) Give mathematical expression for computing the next value of the REAR and FRONT indexes in an array based circular queue.

(b) State situations suitable for a static stack, dynamic array-based stack and linked list-based stack.

(c) What is a priority queue? In case a priority queue is implemented using linked lists, what is the minimum number of lists required for deletion in O(1) time?

(d) How is balance factor computed in an AVL tree? What is its significance?

(e) If the contents of a binary tree be mapped to an array, state the expressions to access the left and right child nodes and the parent node of a node stored at location X.

(f) What is the advantage of threading a binary tree?

(g) How does shell sort improve the performance of Insertion Sort?

(h) What is topological sorting? State at least two of its applications.

(i) Explain the problem of secondary clustering while resolving collisions through quadratic probing.

(j) How is external sorting on tapes different from that on disk? What are the associated issues?

UNIT - I

2. (a) Explain the problem with linear queues. How are these problems overcome with circular queue? (6)

(b) Write a 'C' program to delete a node from a circular linked list. (6.5)

3. (a) Following is 'C' like pseudo code of a function that takes a queue Q as an argument and uses a stack S to do processing. Dry run and explain the result of the function Q2A: (6)

```
void Q2A(Queue *Q) {
   Stack S;  // Create an empty stack S
    while (!isEmpty(Q))
      push(&S, deQueue(Q));
    while (!isEmpty(S))
     enQueue(Q, pop(&S));

}
```

(b) Write a 'C' program to implement multi queue in a single array. (6.5)

UNIT - II

4. (a) How are B-trees different from B^+-trees? What are the merits of B^*-trees over B^+-trees. (6)

   (b) Write 'C' function to delete a node from a tree. (6.5)

5. (a) Demonstrate the insertion of 12, 3, 45, 32, 67, 22 in a post–threaded tree. (6)

   (b) Write 'C' functions for resolving LL and RR imbalance in AVL trees. (6.5)

UNIT - III

6. (a) Write 'C' function to implement DFS on a graph. (6)

   (b) Demonstrate the step-by-step process to sort the keys 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 using quick sort method. (6.5)

7. (a) Explain critical path analysis using an appropriate example. (6)

   (b) Write a 'C' function to implement heap sort. (6.5)

UNIT - IV

8. (a) Write a 'C' function to insert a record in a binary file. (6)

   (b) Demonstrate the running of poly phase merge sort on the keys 78, 45, 6, 78, 90, 23, 43, 45, 67, 80, 21, 44 (6.5)

9. (a) Write the algorithm to generate a new master-file using old master and saved transactions using sequential processing. (6)

   (b) Demonstrate the running of balanced 2-way merge sort using 3 tapes on the keys: 78, 45, 6, 78, 90, 23, 43, 45, 67, 80, 21, 44. (6.5)