FORMAT B
AKGEC/IQAC/QP/04

AJAY KUMAR GARG ENGINEERING COLLEGE, GHAZIABAD

Department of Computer Science and Engineering

Pre-University Test

Course: B. Tech.
Session: 2025-26
Subject: Data Structure
Max Marks: 70

Semester: III
Section: CSE/CSE Allied/IT
Sub. Code: BCS 301
Time: 3 hrs.

OBE Remarks:

Q.No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
CO No.	1	1	2	2	3	4	5	1	2	1	2	3	3	4	4	5	5
Bloom's Level	L2	L1	L1	L2	L4	L3	L1	L5	L3	L4	L6	L2	L3	L4	L4	L3	L4
Weightage CO3: 16						Weightage CO4: 16							Weightage CO5: 16				

Note: Answer **all** the sections.

Section-A

A. Attempt **all** the parts. (7 X 2 =14)

1. How can you represent a sparse matrix in memory?
2. Define and explain the concept of the Time–Space Trade-off in algorithm design.
3. Write the condition for empty and full circular queue.
4. What do you understand by tail recursion?
5. Differentiate between internal sorting and external sorting.
6. In a complete binary tree, if the **number of nodes** is **1012000** find the **height** of the tree.
7. Discuss the different techniques for representing graphs in memory structures.

Section-B

B. Attempt **Any three. (Q. No. 12 is Compulsory)** (3X 7=21)

8. Design and implement a Queue using a linked list in C. Your implementation should include the functions **insert (), delete (),** and **display ().** Further, explain in detail how the implementation of a circular queue effectively overcomes the inherent limitations of a simple linear queue in terms of memory utilization and performance.
9. Write an algorithm to convert a valid arithmetic **infix expression** into an equivalent **postfix expression.** Trace your algorithm for following infix expression.

   A+(B+D)/E-F*(G+H/K).
10. Explain how a polynomial expression can be efficiently represented using a **linked list** data structure. Design and implement a C program to perform addition of two polynomials represented as linked lists, ensuring proper handling of terms with equal powers and different degrees.
11. Explain the structure and characteristics of a **doubly linked list.** Implement an algorithm to insert a new node after a specified node in a singly linked list, clearly describing each step involved in the insertion process.
12. Explain any three commonly used hash functions with suitable examples. Hash function H is defined as **H(key) = key % 9**, with linear probing, and is used to insert the keys **45, 56, 78, 23, 91, 34, 62** into a hash table indexed from **0 to 8.** Determine the final location of the

key 34. Justify your answer, and also calculate the total number of collisions that occur during this probing process.

Section-C

(5 X 7 = 35)

C. Attempt **all** the parts.

13. Attempt any one.

a) Demonstrate the working of the **Merge Sort** algorithm to arrange the following elements in ascending order: **45, 32, 65, 76, 23, 12, 54, 67, 22, 87.** Provide a step-by-step explanation of each phase of the algorithm and analyze its time and space complexity.

b) Explain why the **Quick Sort** algorithm is named **"quick."** Then, demonstrate the step-by-step execution of **Quick Sort** on the following elements, assuming the first element as the **pivot:** **25, 57, 48, 37, 12, 92, 86, 33.**

14. Attempt any one.

a) The order of nodes of a binary tree in **In order** and **Post order** traversal are as follows:
In order : **B, I, D, A, C, G, E, H, F.**
Post order: **I, D, B, G, C, H, F, E, A.**
(i) Draw the corresponding binary tree.
(ii) Write the pre order traversal of the same tree.

b) Explain the concepts of **left-skewed** and **right-skewed binary trees.** Then, **construct an AVL tree** by inserting the given elements in the order of their occurrence.
60,2,14,22,13,111,92,86

15. Attempt any one.

a) Suppose the characters **A, B, C, D, E,** and **F** have the following probabilities:

Character	A	B	C	D	E	F
Probability	0.07	0.09	0.12	0.22	0.23	0.27

i) Construct an **optimal Huffman code** for these characters.
ii) Draw the corresponding **Huffman tree.**
iii) Calculate the **average code length** of the resulting Huffman code.

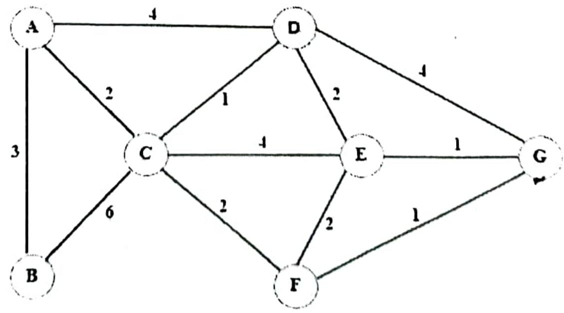
b) Construct a B-Tree of order 5 by inserting the following sequence of keys into an initially empty B-Tree: A, G, F, B, K, D, H, M, J, E, S, I, R, X, C, L, N, T, U, P

i) After constructing the **B-Tree**, delete the keys **J, T,** and **D** in sequence, and draw the resultant **B-Tree** structure.

ii) Explain why the time complexity of the search operation in a **B-Tree** is better than that in a **Binary Search Tree (BST).**

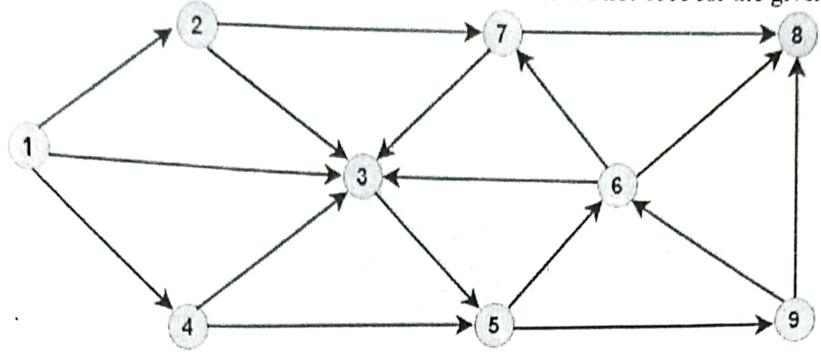
16. Attempt any one.

a) What is a **spanning tree**? Explain its significance in graph theory.



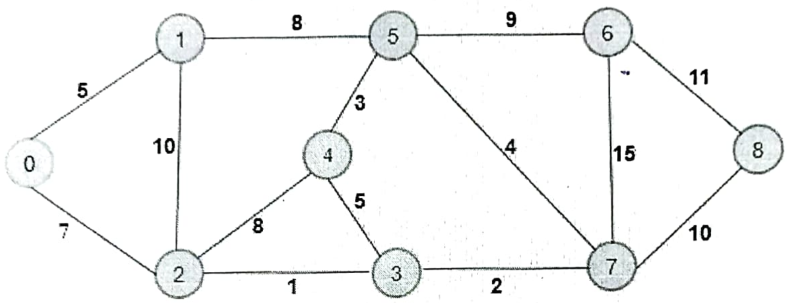
Write and explain **Prim's algorithm** used to obtain a **Minimum Cost Spanning Tree (MCST)** of a connected weighted graph. Apply **Prim's algorithm** to find the **Minimum Cost Spanning Tree** for the given graph.

b) Differentiate between DFS and BFS. Draw the breadth First Tree for the given graph.

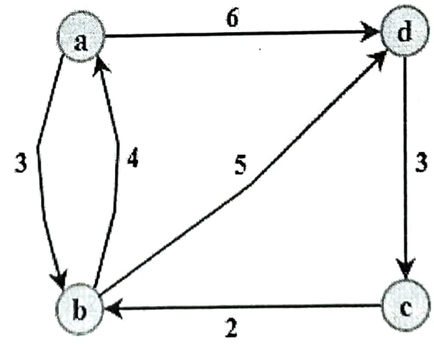


17. Attempt any one.

a) Use **Dijkstra's algorithm** to find the shortest paths from the source vertex **0** to **all other vertices** in the following weighted undirected graph.



b) Write and explain the **Floyd–Warshall algorithm** used to find the **all-pairs shortest paths** in a weighted graph. Using the **Floyd–Warshall algorithm**, determine the **shortest paths among all the vertices** in the given graph.



Faculty Sign

HoD Sign