

- a. Write a pseudocode for the PUSH () and POP () operations using this approach. [7]
- b. Analyse the time complexity of each operation in both best and worst cases. [3]

CO3 BL2

4.a) Given the array: A = [20, 5, 15, 10, 30, 25, 35, 40]. Complete the table given below for the following algorithms: Insertion sort, Counting sort, Selection Sort, Quick sort, Merge sort, Heap Sort and answer the following question:

Algorithm	Time Complexity (Best / Avg / Worst)	Stability	Best Scenario / Suitable For
Insertion Sort			
Selection Sort			
Counting Sort			
Quick Sort			
Merge Sort			
Heap Sort			

- I. Which sorting algorithm would you choose for a small array? Why?
- II. Which algorithm is most efficient for large datasets and why? Which algorithm is suitable for an array with many repeated elements?

OR

4.b) Given the array: A = [12, 7, 14, 10, 9, 20]. Sort the array using Insertion Sort and show the array after each pass. Count the number of comparisons and the number of swaps/assignments.

CO3 BL2

- I. If the array was already sorted, how would the number of comparisons and swaps change? Explain.
- II. If the array was sorted in descending order, how would this affect the Insertion Sort? If the array is very large, which sorting algorithm would you prefer instead of Insertion Sort, and why?

5. Consider the array [15, 10, 20, 8, 12, 16, 25]. Construct a Binary Search Tree (BST) by inserting the elements in the given order. Perform Inorder, Preorder, and Postorder traversals of the constructed BST and write the resulting sequences. Explain how the order in which elements are inserted affects the time complexity of constructing a Binary Search Tree (BST). Give an example of an insertion order that leads to the worst-case construction time and justify why it occurs.

CO4 BL3

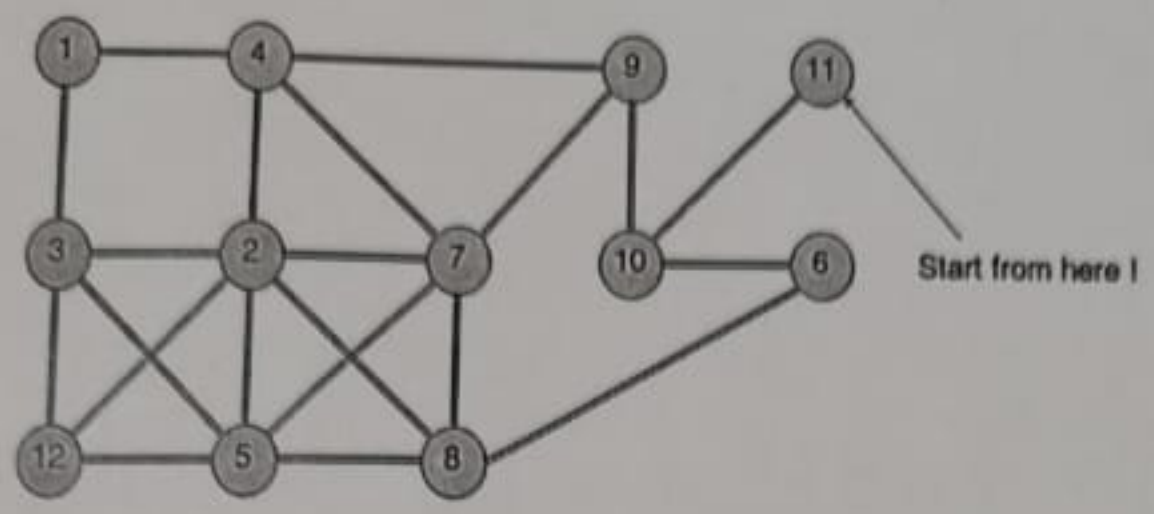
6. Convert the given infix expression $(A + B) * (C \wedge D) / (E + F)$ to postfix notation. Show step-by-step stack operations used in the conversion. (Note: '^' indicates exponential). Assign values to variables as A = 4, B = 8, C = 2, D = 3, E = 5, F = 1. Evaluate the postfix expression step by step using a stack, showing the stack content after each operation. Compute the final result of the expression.

CO4 BL5

7.

Starting from node 11 in an undirected graph, explain how Breadth-First Search (BFS) visits nodes and why it uses a queue. Compare it briefly with Depth-First Search (DFS) in terms of traversal order and data structure.

CO4 BL3



8.

Using the weighted undirected graph below, apply Dijkstra's algorithm to find the shortest distances from vertex A to all other vertices. Show the updates in a **distance table** and briefly explain why Dijkstra's algorithm always finds the shortest paths in graphs with non-negative weights.

CO4 BL3

	A	B	C	D	E
A	0	1	3	0	4
B	1	0	2	5	0
C	3	2	0	6	7
D	0	5	6	0	8
E	4	0	7	8	0

9.

A hash table of size 11 uses **open addressing** with the hash function

CO5 BL5

$$h(k) = (2k + 3) \bmod 11$$

The following keys are to be inserted: **21, 32, 43, 54.**

a) Insert all the keys into the hash table using linear probing to resolve collisions. Show the probe sequence for each key and the final state of the hash table. [6]

b) After the insertions, **the key 43 is deleted**, and a **new key 65 is inserted** using the same method. Show the probe sequence and the resulting table. Finally, explain why deletion in open addressing requires special handling to ensure that subsequent search and insertion operations correctly. [4]

10.

Given the sequence of keys [10, 20, 30, 40, 50, 60, 70], construct a max-heap by inserting them one by one into an initially empty heap, showing all intermediate heap structures and bubble-up (swap) operations; explain why inserting elements in ascending order causes maximum heapification, justify the time complexity of this insertion-based heap construction, and compare it with bottom-up heap construction, stating which method is more efficient and why.

CO5 BL3

⇔⇔⇔ BH/K/TY ⇔⇔⇔

- KEEPING MOBILE PHONE/ANY ELECTRONIC GADGETS, EVEN IN 'OFF' POSITION IS TREATED AS EXAM MALPRACTICE
- DON'T WRITE ANYTHING ON THE QUESTION PAPER

COs	CO Statements
CO1	Understanding the Fundamental Analysis and Time complexity for a given problem.
CO2	Articulate linear, non-linear data structures and legal operations permitted on them.
CO3	Identify and apply suitable algorithms for searching and sorting.
CO4	Discover various tree and graph traversals.
CO5	Explicate hashing, heap and AVL trees and realize their applications.

BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)

Answer ALL Questions
(10 X 10 = 100 Marks)

1.a) Consider the recurrence relation $T(n) = 3T(n/4) + n$, where $T(1) = \Theta(1)$. Using the substitution (iteration) method, expand and simplify the recurrence step by step to derive a tight asymptotic bound for $T(n)$. Clearly show the pattern that emerges during the expansion. Then, illustrate or describe the corresponding recursion tree, indicating the cost at each level and explaining which level contributes most to the overall running time. CO1 BL3

OR

1.b) Compare the recursive and iterative algorithms for computing the n-th Fibonacci number. Derive and discuss their time complexities using asymptotic notation, commenting on best, worst, and average cases. Which method is more efficient overall, and why? CO1 BL3

2. A singly linked list of size n is to be reversed in groups of size k (e.g., reversing every 3 nodes: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$ becomes $3 \rightarrow 2 \rightarrow 1 \rightarrow 6 \rightarrow 5 \rightarrow 4$). CO2 BL3

a) Write a **pseudocode** to perform the group-wise reversal.

b) Derive the worst-case time complexity in terms of n and k . Would this change if the list is doubly linked? Justify.

3. A stack has to be implemented using two queues, Q_1 and Q_2 . For the push operation, the new element is enqueued into Q_1 . For the pop operation, all but the last element is moved from Q_1 to Q_2 ; the last element is dequeued (representing the top of the stack), and then the names of Q_1 and Q_2 are swapped. CO2 BL3