SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur – 603 203

DEPARTMENT OF COMPUTER APPLICATIONS

QUESTION BANK

I SEMESTER

MC4161 -ADVANCED DATA STRUCTURES AND ALGORITHMS

Regulation – 2024

Academic Year 2024-2025 (ODD Semester)

Prepared by

Dr.D.Sridevi

Associate Professor

Department of Computer Applications

SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur – 603 203.

DEPARTMENT OF COMPUTER APPLICATIONS

QUESTION BANK

SUBJECT : MC4161 -ADVANCED DATA STRUCTURES AND ALGORITHMS

SEM / YEAR: I SEM / I Year

	UNIT I - LINEAR DATA STRUCTURES			
	Abstract Data Types (ADTs) – List ADT – Array-Based Implementation – Linked List Implementation Doubly-Linked Lists – Circular Linked Lists – Stack ADT: Implementation of Stacks – Queue ADT: Implementation of Queues – Applications.			
	PART-A			
Q.No.	**Question**	**CO's**	**Level**	**Competence**
1	Define Abstract Data Type. Give any two examples.	CO1	BTL1	Remember
2	Define data structure.	CO1	BTL2	Understand
3	List out the areas in which data structures are applied extensively.	CO1	BTL1	Remember
4	Distinguish between linear and nonlinear data structures.	CO1	BTL1	Remember
5	Compare between linear linked list and circular linked list.	CO1	BTL2	Understand
6	Define an array. Give an example.	CO1	BTL2	Understand
7	List the advantages of linked list.	CO1	BTL1	Remember
8	Examine a doubly linked list with neat diagram.	CO1	BTL1	Remember
9	Interpret the advantages and disadvantages of linked lists over arrays.	CO1	BTL2	Understand
10	Define stack ADT.	CO1	BTL1	Remember
11	Given the prefix for an expression. Write its postfix: **++A*BCD and +*AB*CD**	CO1	BTL2	Understand
12	Given an infix expression convert it into postfix expression usingstack **a+b*(c^d-e)^(f+g*h)-i**	CO1	BTL2	Understand
13	Write the postfix and prefix forms of the expression: **A+B*(C-D)/(P-R)**	CO1	BTL2	Understand
14	Give the purpose of top and pop with suitable example.	CO1	BTL2	Understand
15	What are the two kinds of dequeue?	CO1	BTL2	Understand
16	How to implement stack using linked list?	CO1	BTL2	Understand
17	What are priority queues? What are the ways to implement priorityqueue?	CO1	BTL1	Remember

18	What is circular queue?	CO1	BTL1	Remember
19	Circular queue is better than standard linear queue, why?	CO1	BTL2	Understand
20	Convert **A*(B+D)/E-F*(G+H/K)** into postfix expression.	CO1	BTL2	Understand
21	Show a routine to perform enqueue operation in a queue.	CO1	BTL2	Understand
22	Differentiate between double ended queue and circular queue.	CO1	BTL1	Remember
23	Write an algorithm for deleting an element in a double endedqueue	CO1	BTL1	Remember
24	List any four applications of stack.	CO1	BTL1	Remember

PART-B

Q.No.	Question	Marks	CO's	Level	Competence
1	Write the algorithm for performing operations in a stack. Trace your algorithm with suitable example.	16	CO1	BTL4	Analyze
2	What is Linked List? and write Function to test whether a linked list is empty, Function to test whether current position is the last in a linked list, Function to find the element in the list.	16	CO1	BTL6	Create
3	Develop a C program for linked list implementation of list.	16	CO1	BTL6	Create
4	(i)Explain the list ADT's various operations (Linked and array based) with examples. (ii)Explain the list ADT's various operations (Array based) with examples.	8 8	CO1	BTL3	Apply
5	Analyze and write procedure for circular linked list with create, insert, delete, display operations.	16	CO1	BTL3	Apply
6	Discuss the creation of a doubly linked list and write routine to insert an element in doubly linked list and delete an element in doubly linked list.	16	CO1	BTL4	Analyze
7	Examine the algorithms to implement the doubly linked list and perform all the operations on creating the doubly linked list.	16	CO1	BTL4	Analyze
8	Illustrate the various operation in circular linked lists.	16	CO1	BTL3	Apply
9	(i) Give an algorithm for push and pop operations on stack using a linked list with an example. (ii) Describe the function to examine whether the stack is full () or empty ().	8 8	CO1	BTL4	Analyze
10	(i)Give an algorithm to convert an infix expression to a postfixexpression using stack. (ii)Convert an infix expression to a postfix expression using stack with suitable example.	8 8	CO1	BTL4	Analyze
11	Convert an infix expression to a postfix expression using stack with suitable example.	16	CO1	BTL5	Evaluate
12	(i)Convert an infix expression to a postfix expression using stack for the following expression: ((A + B) – C * (D / E)) + F	8 8	CO1	BTL6	Create

	(ii) Convert an infix expression to a prefix expression using stack for the following expression: ((A + B) – C * (D / E)) + F				
13	Prepare an algorithm to perform the operations in a double ended queue.	16	CO1	BTL5	Evaluate
14	What is a DeQueue? Explain its operation with example?	16	CO1	BTL3	Apply
15	What is a EnQueue? Explain its operation with example?	16	CO1	BTL5	Evaluate
16	Develop a C program for linked list implementation of stack.	16	CO1	BTL4	Analyze
17	What are circular queues? Explain the procedure to insert an element in circular queue and delete an element from a circular queue using array implementation.	16	CO1	BTL5	Evaluate

	UNIT II - ALGORITHMS IN COMPUTING
	Introductions to Algorithms – Iterative and Recursive Algorithms – Designing Algorithms – Analyzing Algorithms – Growth of Functions: Asymptotic Notations – Standard Notations and Common Functions – Recurrences: The Substitution Method – The Recursion – Tree Method
	PART-A

Q.No.	Question	CO's	Level	Competence
1	What do you mean by algorithm?	CO2	BTL1	Remember
2	What is performance measurement?	CO2	BTL2	Understand
3	What are the types of algorithm efficiencies?	CO2	BTL1	Remember
4	What is space complexity?	CO2	BTL1	Remember
5	Define asymptotic notations.	CO2	BTL2	Understand
6	Define the asymptotic notation "Big oh" (0)	CO2	BTL2	Understand
7	Difference between Best Case and Worst-Case Complexities	CO2	BTL1	Remember
8	Define the asymptotic notation "Omega" (Ω).	CO2	BTL1	Remember
9	Define the asymptotic notation "theta" (θ)	CO2	BTL2	Understand
10	Design an algorithm for computing area and circumference of the circle.	CO2	BTL1	Remember
11	How do you measure the efficiency of an algorithm?	CO2	BTL2	Understand
12	Write down the properties of asymptotic notations?	CO2	BTL2	Understand
13	What is a basic operation?	CO2	BTL2	Understand
14	What is validation of algorithm?	CO2	BTL2	Understand
15	What is recursive algorithm?	CO2	BTL2	Understand
16	Define recurrence	CO2	BTL2	Understand

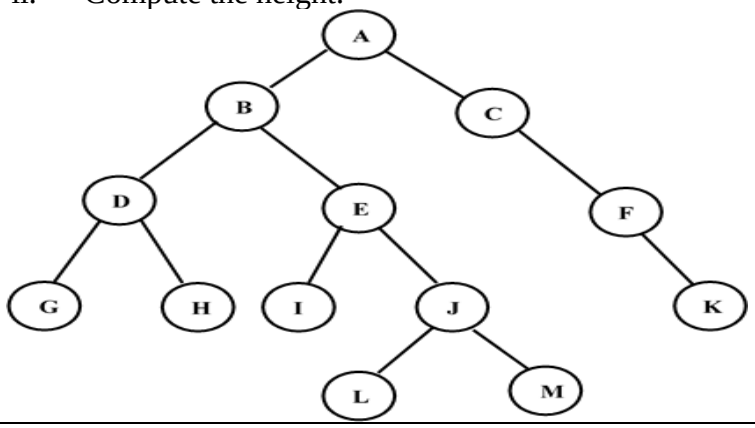
17	Give the general plan for analyzing recursive algorithm	CO2	BTL1	Remember
18	What are all the methods available for solving recurrence relations?	CO2	BTL1	Remember
19	Define Substitution Method	CO2	BTL2	Understand
20	What are the applications of Algorithm Visualization?	CO2	BTL2	Understand
21	List the reasons for choosing an approximate algorithm.	CO2	BTL2	Understand
22	How to measure an algorithm running time?	CO2	BTL1	Remember
23	What are the types of recurrence relations?	CO2	BTL1	Remember
24	What are all the methods available for solving recurrence relations?	CO2	BTL1	Remember

PART-B

Q.No.	Question	Marks	CO's	Level	Competence
1	Discuss briefly the sequence of steps in designing and analyzing an algorithm.	16	CO2	BTL4	Analyze
2	Define the asymptotic notations used for: (i)Best case (ii)Average case and Worst-case analysis with an example.	6 10	CO2	BTL6	Create
3	Explain the general framework for analyzing the efficiency of algorithm	16	CO2	BTL4	Analyze
4	Explain in detail how Big Oh notation used to compare and rank such orders of growth, this notation.	16	CO2	BTL4	Analyze
5	How do you evaluate the performance of the algorithms?	16	CO2	BTL3	Apply
6	Explain in detail how Omega notation used to compare and rank such orders of growth, this notation Ω(big omega)	16	CO2	BTL4	Analyze
7	Explain in detail how Theta notation used to compare and rank such orders of growth, this notation Θ (big theta)	16	CO2	BTL4	Analyze
8	(i) Distinguish between Big Oh, Theta and Omega notation. (ii) Analyze the best, worst and average case analysis for linear search.	8 8	CO2	BTL4	Analyze
9	Illustrate how Time Complexity is calculated. Give an example	16	CO2	BTL3	Apply
10	Illustrate two kinds of efficiency in detail with an example.	16	CO2	BTL3	Apply
11	Illustrate Iterative Method to solve recurrences in detail with an example.	16	CO2	BTL3	Apply
12	(i)Explain Backward Substitution method to solve recurrences in detail with an example. (ii)Explain Forward Substitution method to solve recurrences in detail with an example.	8 8	CO2	BTL4	Analyze
13	Explain Recursion Tree Method to solve recurrences in detail with an example.	16	CO2	BTL4	Analyze

14	Explain Recursion Tree Method to solve recurrences in detail with an example.	16	CO2	BTL4	Analyze
15	Explain various techniques to solve recurrences.	16	CO2	BTL4	Analyze
16	i. Explain how analysis of linear search is done with a suitable illustration. ii. Define recurrence equation and explain how solving recurrence equations are done.	10 6	CO2	BTL3	Apply
17	What is meant by recurrence? Give one example to solve recurrence equations.	16	CO2	BTL3	Apply

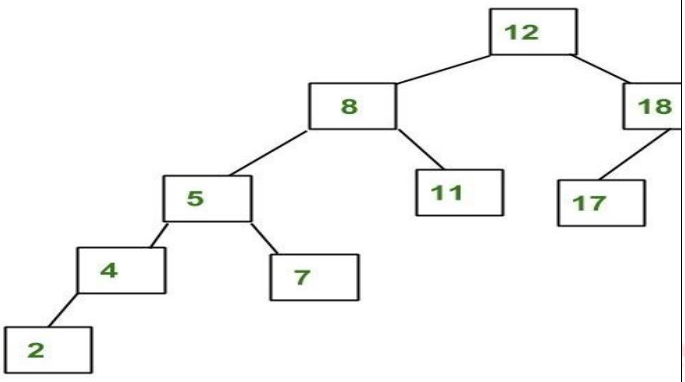
	UNIT III - HIERARCHICAL DATA STRUCTURES & HASHING
	Trees: Preliminaries – Implementation of Trees – Tree Traversals with an Application – Binary Trees: Implementation – Expression Trees – Search Tree ADT – Binary Search Trees – Applications of Trees - Fundamentals of Hashing – Hash Function – Separate Chaining – Open Addressing.
	PART-A

Q.No.	Question	CO's	Level	Competence
1	What is tree traversal?	CO3	BTL 2	Understand
2	Mention the type of traversals.	CO3	BTL 2	Understand
3	Define a binary tree. Give an example.	CO3	BTL 1	Remember
4	Create an expression tree for the expression. **4+((7+9) *2).**	CO3	BTL 2	Understand
5	Differentiate AVL tree and Binary search tree.	CO3	BTL 1	Remember
6	Give the various types of rotations in AVL tree during the insertion of a node?	CO3	BTL 1	Remember
7	For the given tree i. List the siblings for node E ii. Compute the height. 	CO3	BTL 2	Understand
8	List the steps in deleting a node from a binary search tree.	CO3	BTL 1	Remember
9	Define the balance factor of AVL Tree.	CO3	BTL 1	Remember

10	Define Separate Chaining.	CO3	BTL 2	Understand
11	Define an expression tree. Give an example of it.	CO3	BTL 1	Remember
12	Define AVL Tree.	CO3	BTL 1	Remember
13	Identify the properties of Binary Search Tree.	CO3	BTL 1	Remember
14	Identify the type of traversal that gives the data in ascending order.	CO3	BTL2	Understand
15	Identify the properties of AVL Tree.	CO3	BTL2	Understand
16	List out the various operations that can be performed on Binary Search Tree to make it a height balanced one.	CO3	BTL 1	Remember
17	Identify the three cases for deleting a node from a binary search tree.	CO3	BTL 1	Remember
18	Define leaves in a tree.	CO3	BTL1	Remember
19	What do you mean by level of the tree?	CO3	BTL2	Understand
20	Identify the properties of a binary tree.	CO3	BTL1	Remember
21	Define Hashing.	CO3	BTL2	Understand
22	List the Different ways of Open Addressing.	CO3	BTL1	Remember
23	List out the applications of Trees.	CO3	BTL2	Understand
24	What do you mean by Heap?	CO3	BTL1	Remember

PART-B

Q.No.	Question	Marks	CO's	Level	Competence
1	Illustrate an algorithm for traversals of a binary tree with an example.	16	CO3	BTL3	Apply
2	Explain the following operations on a binary search tree with suitable algorithm: i. Find a node ii. Find minimum and maximum elements of BST.	6 10	CO3	BTL3	Apply
3	Write an algorithm for inserting nodes in a binary search tree and explain with an example.	16	CO3	BTL3	Apply
4	Describe the various operations that can be performed on a Binary Search Tree with example for each operation.	16	CO3	BTL3	Apply
5	i. Discuss in detail the various methods in which a binary tree can be represented. ii. Discuss the advantage and disadvantage of each method.	10 6	CO3	BTL4	Analyze

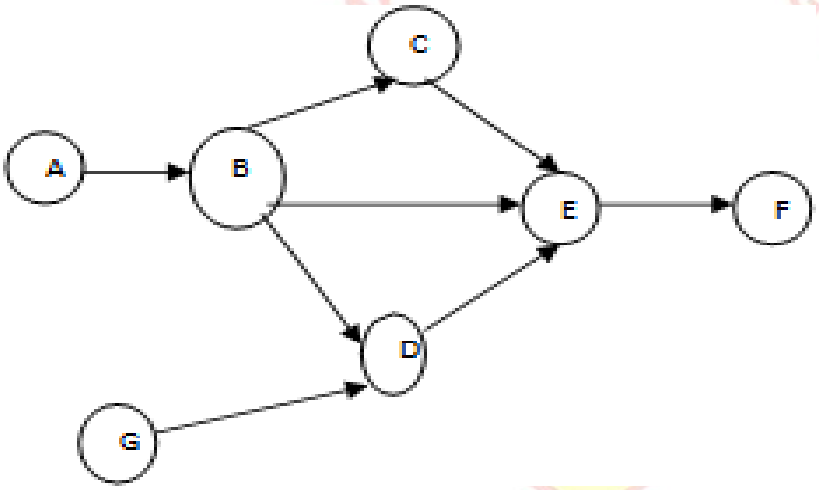
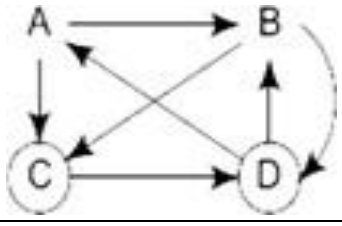
6	i. Create a binary search tree using the following data elements **45, 39, 56, 12, 34, 78, 32, 10, 89, 54, 67, 81** ii. Explain the steps to convert general tree to binary tree?	10 6	CO3	BTL5	Evaluate
7	Describe the three cases for deleting a node from a binary search tree with an example for each case.	16	CO3	BTL5	Evaluate
8	i. Discuss whether the following tree is AVL tree or not. If yes explain, how is it balanced and if not balance the tree  ii. Explain how deletion can take place in AVL trees with suitable algorithm.	10 6	CO3	BTL6	Create
9	i. What are AVL trees? Describe the different rotations defined for AVL tree. ii. Insert the following elements step by step in sequence into an empty AVL tree 63, 9, 19, 27, 18, 108, 99, 81.	8 8	CO3	BTL5	Evaluate
10	Develop a routine for post order traversal. Is it possible to find minimum and maximum value in the binary search tree using traversals? Discuss in detail with an example.	16	CO3	BTL3	Apply
11	Write an algorithm for deleting a node in a binary search tree considering the different cases of a node to be deleted.	16	CO3	BTL3	Apply
12	Explain the construction of expression tree with example. Give the applications of trees.	16	CO3	BTL4	Analyze
13	Explain Deletion in Binary tree by merging and copying.	16	CO3	BTL3	Apply
14	i. Explain how to delete an element from the binary search tree. ii. Write recursive algorithm for pre order traversal.	8 8	CO3	BTL3	Apply
15	Construct an expression tree for the expression (a+b*c) + ((d*e+f)*g). Give the outputs when you apply inorder, preorder and postorder traversals.	16	CO3	BTL3	Apply
16	Show each step of hash table entries for the given data set using linear probing 12,45,67,88,27,78,20,62,36,55 (size=10)	16	CO3	BTL6	Create
17	Apply quadratic hashing to fill the hash table of size 11 elements 20,5,10,22,33,40,50,30,51,31	16	CO3	BTL6	Create

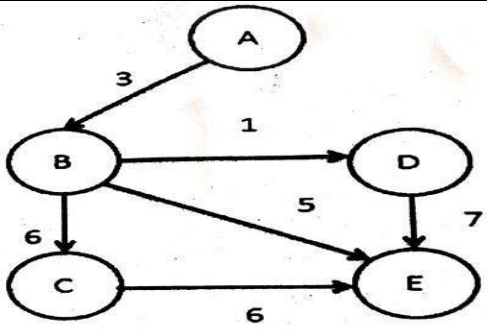
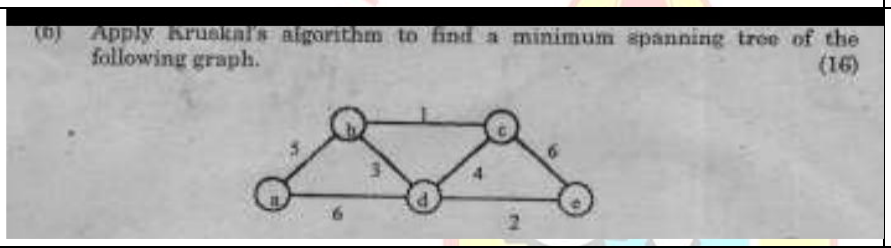
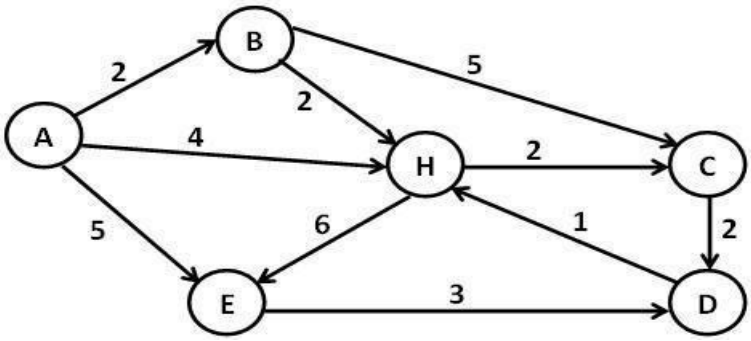
	UNIT IV - SORTING AND GRAPHS
	Sorting Algorithms: Insertion Sort, Quick Sort, Heap Sort - Graphs: Representation of Graphs – Graph Traversals – Topological Sort – Shortest Path Algorithms: Dijkstra's Algorithm – Minimum Spanning Tree: Prim's and Kruskal's Algorithm.
	PART-A

Q.No.	Question	CO's	Level	Competence
1	Define Heap sort?	CO4	BTL 1	Remember
2	Define Quick Sort.	CO4	BTL 2	Understand
3	What is the difference between quicksort and merge sort?	CO4	BTL 2	Understand
4	Define graph.	CO4	BTL 2	Understand
5	Create an undirected graph and its adjacency matrix for the following specification of a graph G. V(G)=1,2,3,4; E(G) ={ (1,2),(1,3),(3,3),3,4),(4,1)}	CO4	BTL 2	Understand
6	What is meant by bi-connected graph?	CO4	BTL 1	Remember
7	Give the purpose of Dijkstra's algorithm.	CO4	BTL 2	Understand
8	Differentiate cyclic and acyclic graph	CO4	BTL 1	Remember
9	Classify strongly connected and weakly connected graph.	CO4	BTL 1	Remember
10	What is an articulation point? Give example.	CO4	BTL 1	Remember
11	What are the representations of the graph?	CO4	BTL 1	Remember
12	Define minimum spanning tree. Give an example	CO4	BTL 1	Remember
13	State the principle of Topological sorting.	CO4	BTL 1	Remember
14	Explain procedure for Depth first search algorithm.	CO4	BTL 2	Understand
15	What is Dynamic programming technique? Explain it with an example.	CO4	BTL 2	Understand
16	Prove that the number of edges in a complete graph of n vertices in n (n-1)/2.	CO4	BTL 2	Understand
17	Assess the minimum number spanning tree possible for a complete graph with n vertices.	CO4	BTL 2	Understand
18	Give two applications of graphs.	CO4	BTL 2	Understand
19	What is visiting and traversing in a graph?	CO4	BTL 1	Remember
20	When is a graph said to be weakly connected?	CO4	BTL 2	Understand
21	What is Greedy method? Give an example.	CO4	BTL 1	Remember
22	What is in-degree of a graph. Give suitable example.	CO4	BTL 2	Understand
23	What is meant by out-degree of a graph?	CO4	BTL 1	Remember

24	What are the two basic parts of Heap sort?	CO4	BTL2	Understand

PART-B

Q.No.	Questions	Marks	CO's	Level	Competence
1	Explain in detail in merge sort give an example	16	CO4	BTL4	Analyze
2	Distinguish between Quick sort and Merge sort, and arrange the following numbers in increasing order using merge sort. (18, 29, 68, 32, 43, 37, 87, 24, 47, 50)	16	CO4	BTL3	Apply
3	Sort the following set of elements using merge sort: 12, 24, 8, 71, 4, 23, 6, 89, and 56.	16	CO4	BTL3	Apply
4	Explain the concept of heap sort with example?	16	CO4	BTL4	Analyze
5	Describe in detail about the following representations of a graph. i. Adjacency Matrix ii. Adjacency List	8 8	CO4	BTL5	Evaluate
6	i) Consider the given directed acyclic graph D. Sort the nodes D by applying topological sort on 'D'.  ii.Consider the graph given below and show its adjacency list in the memory. iii.  	8 8	CO4	BTL6	Create
7	Describe an appropriate algorithm to find the shortest path from 'A' to every other node of A for the given graph.	16	CO4	BTL4	Analyze

					
8	i. Examine topological sorting of a graph G with suitable example. i. Explain Dynamic programming with suitable examples.	8 8	CO4	BTL3	Apply
9	Differentiate depth-first search and breadth-first search traversal of a graph with suitable examples.	16	CO4	BTL3	Apply
10	i. Explain with algorithm, How DFS be performed on an undirected graph. ii. Show the algorithm for finding connected components of an undirected graph using DFS, and derive the time complexity of the algorithm.	10 6	CO4	BTL4	Analyze
11	Discuss an algorithm for Breadth first Search on a graph. Give an example based on the algorithm.	10 6	CO4	BTL4	Analyze
12		16	CO4	BTL4	Analyze
13	Develop an algorithm to compute the shortest path using Dijkstra's algorithm. Validate the algorithm with suitable example.	16	CO4	BTL6	Create
14	Explain the depth first approach of finding articulation points in a connected graph with necessary algorithm.	16	CO4	BTL6	Create
15	i. Write short notes on Bi-connectivity. ii. Express different types of graphs with example.	8 8	CO4	BTL3	Apply
16	Explain the various applications of graphs.	16	CO4	BTL3	Apply
17	Using Dijkstra's algorithm to find the shortest path from the source node A. 	16	CO4	Create	BTL6

	UNIT V - ALGORITHM DESIGN TECHNIQUES			
	Greedy Algorithms: Huffman Codes – Divide and Conquer: Merge Sort – Dynamic Programming: Using a Table instead of Recursion – Ordering Matrix Multiplications – Introduction to NP Completeness			
	PART-A			
Q.No.	**Question**	**CO's**	**Level**	**Competence**
1	Mention different algorithm design techniques.	CO5	BTL 1	Remember
2	Mention the two properties of sorting algorithms.	CO5	BTL 2	Understand
3	State greedy technique.	CO5	BTL 2	Understand
4	Define dynamic programming.	CO5	BTL 2	Understand
5	Define divide and conquer design technique	CO5	BTL 2	Understand
6	State the Principle of Optimality.	CO5	BTL 1	Remember
7	What is Huffman trees?	CO5	BTL 2	Understand
8	List the advantage of Huffman's encoding?	CO5	BTL 1	Remember
9	What do you mean by Huffman code?	CO5	BTL 1	Remember
10	What is greedy method?	CO5	BTL 1	Remember
11	What do you mean by row major and column major?	CO5	BTL 1	Remember
12	Show the general procedure of dynamic programming.	CO5	BTL 1	Remember
13	Define Kruskal Algorithm.	CO5	BTL 1	Remember
14	List the features of dynamic programming?	CO5	BTL 2	Understand
15	How Dynamic Programming is used to solve Knapsack Problem?	CO5	BTL 2	Understand
16	List two major parts in Huffman Coding.	CO5	BTL 2	Understand
17	Define NP Problem.	CO5	BTL 2	Understand
18	Differentiate P and NP.	CO5	BTL 2	Understand
19	How NP-hard problems are different from NP-Complete?	CO5	BTL 1	Remember
20	Give some examples of non-polynomial problem.	CO5	BTL 2	Understand
21	Give some examples of Polynomial problem.	CO5	BTL 1	Remember
22	Define non-polynomial problem.	CO5	BTL 2	Understand
23	Define Polynomial (P) problem.	CO5	BTL 1	Remember
24	On what basis problems are classified?	CO5	BTL2	Understand

PART-B

Q.No.	Question	Marks	CO's	Level	Competence
1	Discuss briefly the sequence of steps in designing and analyzing an algorithm.	16	CO5	BTL4	Analyze
2	Explain in detail about divide and conquer approach.	16	CO5	BTL4	Analyze
3	Describe in detail about merge sort with an example.	16	CO5	BTL4	Analyze
4	Explain in detail about Greedy Algorithms.	16	CO5	BTL4	Analyze
5	Explain in detail about Dynamic Programming.	16	CO5	BTL4	Analyze
6	Explain Kruskal's Algorithm.	16	CO5	BTL4	Analyze
7	Illustrate Prim's Algorithm in detail with an example.	16	CO5	BTL3	Apply
8	Illustrate Greedy Method with an example.	16	CO5	BTL3	Apply
9	Explain in detail that what does dynamic programming have in common with divide-and-Conquer?	16	CO5	BTL4	Analyze
10	Explain how Floyd's Algorithm works.	16	CO5	BTL4	Analyze
11	Illustrate Huffman code algorithm and derive its complexity.	16	CO5	BTL3	Apply
12	Outline the Dynamic Programming approach to solve the Optimal Binary Search Tree problem and analyze it time complexity.	16	CO5	BTL4	Analyze
13	Explain in detail two major parts in Huffman Coding.	16	CO5	BTL4	Analyze
14	Explain Steps to build Huffman Tree with an example.	16	CO5	BTL4	Analyze
15	Explain how Matrix – chain Multiplication problem can be solved using dynamic programming with suitable example.	16	CO5	BTL4	Analyze
16	Describe in detail about P and NP problems.	16	CO5	BTL4	Analyze
17	Illustrate NP hard problems in detail with an example.	16	CO5	BTL3	Apply