CBCS SCHEME

USN | 4 | 1 | | | | | | | | BCS401

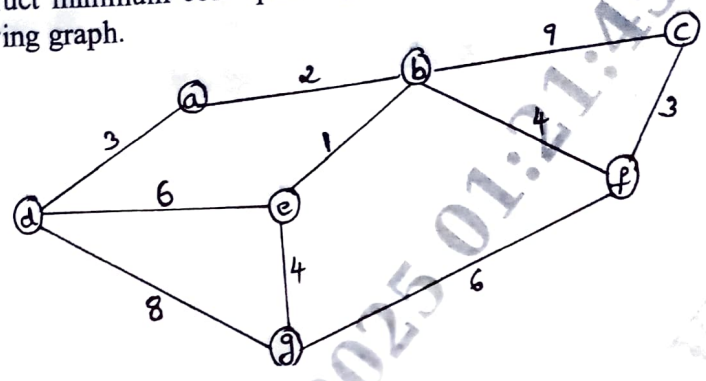
Fourth Semester B.E./B.Tech. Degree Examination, June/July 2025
Analysis and Design of Algorithms

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

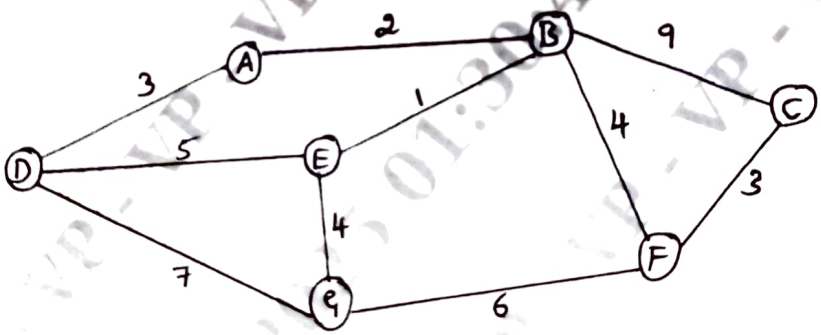
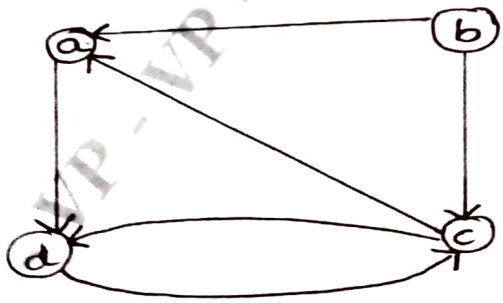
		Module – 1	M	L	C
Q.1	a.	Define algorithm Explain asymptotic notations Bigh oh, Big omega and Big theta notations.	08	L2	CO1
	b.	Explain the general plan for analyzing the efficiency of a recursive algorithm. Suggest a recursive algorithm to find factorial of number. Derive its efficiency.	08	L3	CO1
	c.	If $t_1(n) \in O(g_1(n))$ and $t_2(n) \in O(g_2(n))$ then show that $t_1(n) + t_2(n) \in o(\max\{g_1(n), g_2(n)\})$	04	L2	CO1
		OR			
Q.2	a.	With a neat diagram explain different steps in designing and analyzing algorithm.	08	L2	CO1
	b.	Write an algorithm to find the max element in an array of n elements. Give the mathematical analysis of this non- recursive algorithm.	08	L3	CO1
	c.	With the algorithm derive the worst case efficiency for selection sort.	04	L3	CO1
		Module – 2			
Q.3	a.	Explain the concept of divide and conquer. Design an algorithm for merge sort and derive its time complexity.	10	L3	CO2
	b.	Design an algorithm for insertion algorithm and obtain its time complexity. Apply insertion sort on these elements. 89, 45, 68, 90, 29, 34, 17	10	L3	CO2
		OR			
Q.4	a.	Design an algorithm for Quick sort. Apply quick sort on these elements. 5, 3, 1, 9, 8, 2, 4, 7.	10	L3	CO2
	b.	Explain Strassen's Matrix multiplication and derive its time complexity.	10	L2	CO2
		Module – 3			
Q.5	a.	Define AVL trees. Explain its four rotation types.	10	L2	CO3
	b.	Design an algorithm for Heap sort. Construct bottom – up heap for the list 15, 19, 10, 7, 17, 16.	10	L3	CO4
		OR			
Q.6	a.	Design Horspool's Algorithm for string matching Apply Horspool algorithm to find pattern BARBER in the test: JIM_SAW_ME_IN_A_BARBERSHOP.	10	L3	CO4
	b.	Define heap. Explain the properties of heap along with its representation.	10	L2	CO3

Module – 4

Q.7	a.	Construct minimum cost spanning tree using Kruskal's algorithm for the following graph.  Fig. 7(a)	10	L3	CO4
	b.	What are Huffman trees? Construct the Huffman tree for the following data i) Encode the text ABAC ABAD ii) Decode the code100010111001010	10	L3	CO4

Character	A	B	C	D	-
Probability	0.4	0.1	0.2	0.15	0.15

OR

Q.8	a.	Apply Dijkstra's algorithm to fine single source shortest path for the given graph by considering A as the source vertex.  Fig.8 (a)	10	L3	CO4
	b.	Define transitive closure of a graph. Apply Warshall's algorithm to compute transitive closure of a directed graph.  Fig.8 (b)	10	L3	CO4

Module – 5

<table>
<tr><td>Q.9</td><td>a.</td><td>Explain the following with examples.
i) P problem
ii) NP problem
ii) NP-Complete problem
iv) NP – Hard problem</td><td>10</td><td>L2</td><td>CO5</td></tr>
<tr><td></td><td>b.</td><td>What is backtracking? Apply backtracking to solve the below instance of sum of subset problem.
S = { 1, 2, 5, 6, 8} and d = 9.</td><td>10</td><td>L3</td><td>CO6</td></tr>
<tr><td colspan="6" align="center">OR</td></tr>
<tr><td>Q.10</td><td>a.</td><td>Illustrate N Queen's problem using backtracking to solve 4 – Queens problem.</td><td>10</td><td>L2</td><td>CO6</td></tr>
<tr><td></td><td>b.</td><td>Using Branch and Bound method solve the below instance of Knapsack Problem.
<table><tr><td>Item</td><td>Weight</td><td>Value</td></tr><tr><td>1</td><td>4</td><td>40</td></tr><tr><td>2</td><td>7</td><td>42</td></tr><tr><td>3</td><td>5</td><td>25</td></tr><tr><td>4</td><td>3</td><td>12</td></tr></table>Capacity = 10</td><td>10</td><td>L3</td><td>CO6</td></tr>
</table>

* * * * *