U.S.N.										

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

June 2025 Semester End Main Examinations

Programme: B.E. | **Semester: IV**

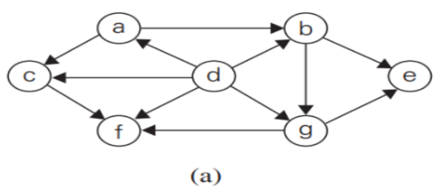
Branch: Computer Science and Engineering | **Duration: 3 hrs.**

Course Code: 23CS4PCADA / 22CS4PCADA / 19CS4PCADA | **Max Marks: 100**

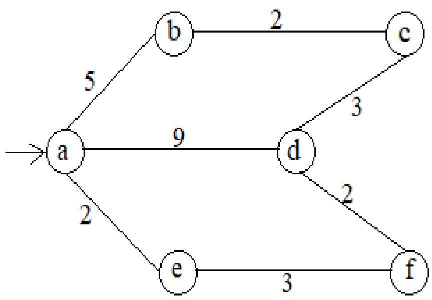
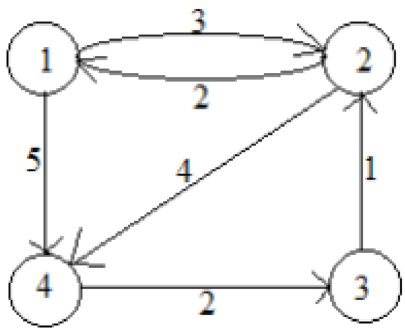
Course: Analysis and Design of Algorithms

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

		UNIT – I	*CO*	*PO*	Marks
1	a)	List and identify the fundamental properties of algorithmic problem solving.	*CO2*	*PO1*	**8**
	b)	Explain different asymptotic notations used to represent the time complexities.	*CO1*	*PO2*	**6**
	c)	Explain the general plan for analyzing the efficiency of a non-recursive algorithm with an example	*CO1*	*PO2*	**6**
		OR			
2	a)	Compare and analyze the time and space complexity of recursive and non-recursive versions of the algorithm to calculate the factorial of a number. Discuss the advantages and disadvantages of each approach.	*CO1*	*PO2*	**8**
	b)	Using Master theorem obtain the time complexity of the following recurrence relations i) $T(n) = 16T(n/4) + n$ ii) $T(n) = 3T(n/3) + n/2$ iii) $T(n) = 6T(n/3) + n^2 \log n$	*CO1*	*PO2*	**6**
	c)	Explain the general plan for analyzing the efficiency of a recursive algorithm with an example.	*CO1*	*PO2*	**6**
		UNIT – II			
3	a)	Build a pattern matching algorithm to search a given pattern using brute force technique and outline the time complexity for the same	*CO2*	*PO1*	**7**
	b)	i) Using decrease and conquer technique obtain the topological sorting for the following digraph (a)  (a)	*CO2*	*PO1*	**7**

<table>
<tr><td></td><td></td><td></td><td>ii)The table shows the courses with their prerequisites. Find the correct order in which a student can register for all the courses.
<table><tr><td>Prerequisite</td><td>Course</td></tr><tr><td>Mathematics</td><td>Physics</td></tr><tr><td>Mathematics</td><td>CS</td></tr><tr><td>CS</td><td>AI</td></tr><tr><td>Physics</td><td>AI</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>c)</td><td>Compute the product of 50*65 using Russian peasant method.</td><td>CO2</td><td>PO1</td><td>6</td></tr>
<tr><td></td><td></td><td></td><td>OR</td><td></td><td></td><td></td></tr>
<tr><td></td><td>4</td><td>a)</td><td>Obtain the optimal solution for the knapsack problem using exhaustive search. Given, knapsack capacity M = 40, number of items, n = 4, {w1, w2, w3, w4} = {20, 25, 10, 15} represents weights of 4 objects, {p1, p2, p3, p4} ={30, 40, 35, 10} represents profits of 4 objects.</td><td>CO2</td><td>PO1</td><td>6</td></tr>
<tr><td></td><td></td><td>b)</td><td>Write the Johnson Trotter algorithm and demonstrate the same for the objects 2 ,4 ,6 ,8.</td><td>CO2</td><td>PO1</td><td>7</td></tr>
<tr><td></td><td></td><td>c)</td><td>Apply the partition-based algorithm to find the median of the following list of nine elements :4,1,10,8,7,12,9,2,15.</td><td>CO2</td><td>PO1</td><td>7</td></tr>
<tr><td></td><td></td><td></td><td>UNIT - III</td><td></td><td></td><td></td></tr>
<tr><td></td><td>5</td><td>a)</td><td>A search engine retrieves 10 webpages based on a user query and assigns them relevance scores as [85, 60, 95, 70, 50, 90, 80, 75, 65, 55]. Apply the fastest sorting algorithm to sort the pages based on scores by choosing pivot element as 55.</td><td>CO2</td><td>PO1</td><td>8</td></tr>
<tr><td></td><td></td><td>b)</td><td>You are developing a plagiarism detection system that needs to quickly identify if a specific sequence of words (pattern) appears within a large text (document). To improve performance, your system uses the Boyer-Moore string matching algorithm.
Given:
Text: "The quick brown fox jumps over the lazy dog. The quick brown fox is clever."
Pattern: "quick brown"</td><td>CO2</td><td>PO1</td><td>8</td></tr>
<tr><td></td><td></td><td>c)</td><td>With an example explain Horner’s rule.</td><td>CO2</td><td>PO1</td><td>4</td></tr>
<tr><td></td><td></td><td></td><td>OR</td><td></td><td></td><td></td></tr>
<tr><td></td><td>6</td><td>a)</td><td>You are given a list of job priorities to be executed by a server: [12, 7, 6, 10, 8, 20, 15, 2]. Using a transform and conquer technique, sort the jobs in descending order of priority.</td><td>CO2</td><td>PO1</td><td>8</td></tr>
<tr><td></td><td></td><td>b)</td><td>Assume You are analyzing a DNA strand represented as a string:
Text: "AGCTTAGCTAAGCTTGCAGCTT"
Pattern: "GCTT"
Apply an appropriate algorithm to find the number of comparisons made and the shift values used during the search.</td><td>CO2</td><td>PO1</td><td>8</td></tr>
</table>

<table>
<tr><td></td><td></td><td>c)</td><td>Multiply the matrices shown below using Strassen's matrix multiplication.
$\begin{bmatrix}21 & 0\\ 0 & 1\end{bmatrix}$ $\begin{bmatrix}11 & 0\\ 0 & 30\end{bmatrix}$</td><td>CO2</td><td>PO1</td><td>4</td></tr>
<tr><td></td><td></td><td></td><td>UNIT - IV</td><td></td><td></td><td></td></tr>
<tr><td></td><td>7</td><td>a)</td><td>A company is analyzing its internal communication network across different departments. There are 5 departments labeled A, B, C, D, and E. The communication between them is represented as a directed graph, where an edge from department X to Y means information can be sent directly from X to Y. Compute the transitive closure of the communication matrix.
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr>
<tr><td>A</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>B</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>C</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>E</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table></td><td>CO2</td><td>PO1</td><td>10</td></tr>
<tr><td></td><td></td><td>b)</td><td>Construct a Dijkstra's algorithm and find the Single Source Shortest Path (SSSP) for the following graph by taking a as source vertex.

</td><td>CO2</td><td>PO1</td><td>10</td></tr>
<tr><td></td><td></td><td></td><td>OR</td><td></td><td></td><td></td></tr>
<tr><td></td><td>8</td><td>a)</td><td>Develop an algorithm to find all pair shortest path. Apply the algorithm to find the all-pair shortest path of a below shown graph using dynamic programming.

</td><td>CO2</td><td>PO1</td><td>10</td></tr>
</table>

<table>
<tr><td></td><td></td><td>b)</td><td>A messaging app stores common phrases as compressed text using Huffman encoding. The following phrases occur with the frequencies listed:

Phrase Frequency
"OK" 100
"Thanks" 40
"LOL" 30
"See you" 20
"On my way" 10

Construct the Huffman Tree for the above data and generate the Huffman codes for each phrase. Also find the storage space required before and after coding.</td><td>CO2</td><td>PO1</td><td>10</td></tr>
<tr><td></td><td></td><td></td><td>UNIT - V</td><td></td><td></td><td></td></tr>
<tr><td></td><td>9</td><td>a)</td><td>Illustrate with an example how backtracking technique is used in N-Queens problem for N=4.</td><td>CO2</td><td>PO1</td><td>6</td></tr>
<tr><td></td><td></td><td>b)</td><td>With an example explain NP hard and NP complete problems</td><td>CO3</td><td>PO1</td><td>8</td></tr>
<tr><td></td><td></td><td>c)</td><td>A surveillance system needs to place cameras at intersections in a city such that every road is monitored by at least one camera at its endpoints. The city’s map is represented as an undirected graph, where intersections are vertices and roads are edges:

Vertices (Intersections): {A, B, C, D, E}

Edges (Roads): {(A, B), (A, C), (B, D), (C, D), (D, E)}

Model the above scenario as vertex cover problem and determine whether there is a vertex cover of size 3.</td><td>CO3</td><td>PO1</td><td>6</td></tr>
<tr><td></td><td></td><td></td><td>OR</td><td></td><td></td><td></td></tr>
<tr><td></td><td>10</td><td>a)</td><td>Examine for the 0 /1 Knapsack problem using Branch and Bound where Knapsack capacity m=15 and n=4

<table><tr><td>Profit</td><td>10</td><td>10</td><td>12</td><td>18</td></tr><tr><td>Weight</td><td>2</td><td>4</td><td>6</td><td>9</td></tr></table></td><td>CO2</td><td>PO1</td><td>10</td></tr>
<tr><td></td><td></td><td>b)</td><td>What is backtracking? Apply backtracking to solve the below instance of sum of subset problem S =5,10,12,13,15,18}, d=30. Write the appropriate steps to explain the iterations.</td><td>CO2</td><td>PO1</td><td>10</td></tr>
</table>

******