U.S.N.										

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

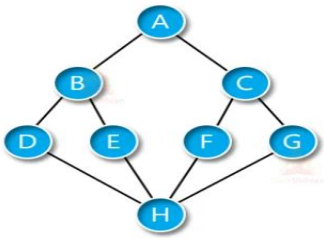
Autonomous Institute Affiliated to VTU

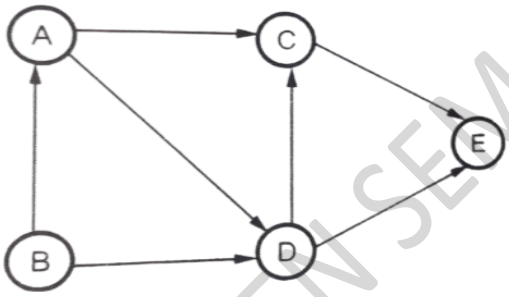
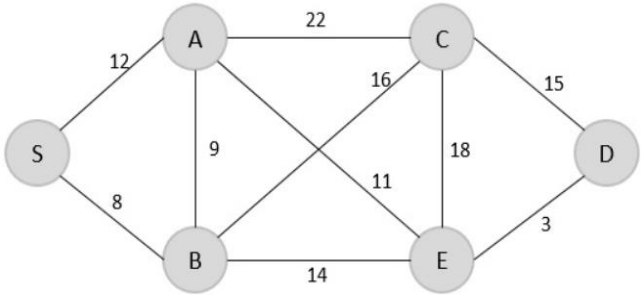
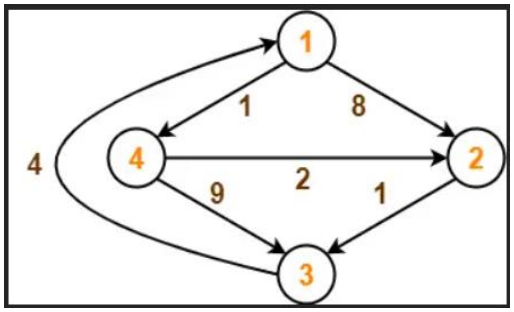
June 2025 Semester End Main Examinations

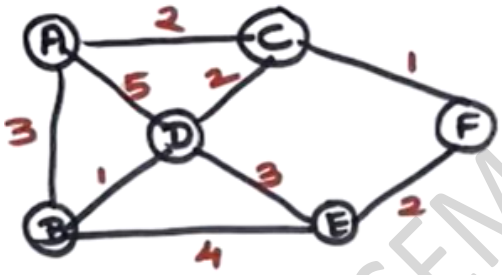
Programme: B.E. **Semester: IV**
Branch: Information Science and Engineering **Duration: 3 hrs.**
Course Code: 23IS4PCADA/22IS4PCADA/19IS4PCADA **Max Marks: 100**
Course: Analysis and Design of Algorithms

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)	Illustrate the asymptotic notations to measure the efficiency of an algorithm.	CO1	PO1	4
	b)	Apply Selection Sort for the given list: {23, 78, 45, 8, 32, 56} and derive its efficiency.	CO2	PO2	8
	c)	Explain general plan for analyzing efficiency of recursive algorithms and apply the same to find time complexity of the Tower of Hanoi Algorithm.	CO2	PO2	8
		OR			
2	a)	Derive the time complexity to find the n^{th} Fibonacci number.	CO2	PO2	4
	b)	Write the bubble sort algorithm. Apply the same to sort the list of the given elements {5, 3, 8, 4, 6} in ascending order.	CO2	PO1	8
	c)	Provide an algorithm to find the uniqueness of elements in an array and prove its efficiency as $\Theta(n^2)$.	CO2	PO2	8
		UNIT - II			
3	a)	Write the quick sort algorithm. Apply the algorithm to sort the list {5, 3, 1, 9, 8, 2, 4, 7}.	CO1	PO1	10
	b)	Find the BFS traversal starting from vertex A for the following Graph. Also, provide the BFS Algorithm. 	CO1	PO1	10

			OR			
	4	a)	Discuss how Strassen's Matrix Multiplication saves time when used with divide and conquer, with its equations. Apply Strassen's Matrix Multiplication for the following: $\begin{bmatrix} 1 & 0 & 2 & 1 \\ 4 & 1 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 5 & 0 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 & 1 \\ 2 & 1 & 0 & 4 \\ 2 & 0 & 1 & 1 \\ 1 & 3 & 5 & 0 \end{bmatrix}$	***CO1***	***PO1***	**10**
		b)	Obtain a topological sorting algorithm using the Decrease and Conquer technique. Apply the same method to solve the following topological sorting problem. 	***CO1***	***PO1***	**10**
			UNIT - III			
	5	a)	Find the minimum spanning tree using Prim's algorithm for the graph given below with S as the arbitrary root. 	***CO1***	***PO1***	**6**
		b)	Write Floyd's algorithm and apply the same to solve the following problem. 	***CO1***	***PO1***	**8**

		c)	Apply Knapsack algorithm using dynamic programming to find an optimal solution for the given instances: (Capacity of the Knapsack =8). **Item** | **A** | **B** | **C** | **D** **Profit** | 2 | 4 | 7 | 10 **Weight** | 1 | 3 | 5 | 7	*CO1*	*PO1*	**6**
			OR			
	6	a)	Provide single-source shortest path algorithm. Apply the algorithm on the following graph, with vertex "A" as the source.  	*CO1*	*PO1*	**6**
		b)	Construct the Huffman code for the following data. symbol | A | B | C | D | _ frequency | 0.35 | 0.1 | 0.2 | 0.2 | 0.15 Also i) encode DAD and ADD ii) Decode 10011011011101	*CO1*	*PO1*	**8**
		c)	Write the algorithm for computing binomial coefficient C(n, k) using dynamic programming approach. Represent the binomial coefficient table for C(7, 5).	*CO1*	*PO1*	**6**
			UNIT - IV			
	7	a)	For the input {30, 20, 56, 75, 31, 19} and hash function h(k)=k mod 11 i) Construct the closed hash table ii) Find the largest and average number of key comparisons in a successful search for hash table	*CO1*	*PO2*	**10**
		b)	Sort the following list of elements {82, 90, 10, 12, 15, 77, 55, 23} by using Heap sort technique in an ascending order. Also, write an algorithm for the same.	*CO1*	*PO1*	**10**
			OR			
	8	a)	Provide Boyer-Moore matcher algorithm. Apply the same algorithm to search for the pattern **STOR** in the text **STOU TOR STOR.**	*CO1*	*PO1*	**7**
		b)	Construct an AVL Tree for the following set of elements: {1, 2, 3, 4, 5, 6, 7}	*CO1*	*PO1*	**6**

<table>
<tr><td></td><td></td><td>c)</td><td>Design a Presorting-based algorithm to check element uniqueness in an array and determine its time efficiency.</td><td>CO2</td><td>PO2</td><td>7</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>Apply the Backtracking technique to solve the following instance of the subset sum problem S = {3, 5, 6, 7} and d=15. Also, represent the State Space Tree for the above Problem</td><td>CO1</td><td>PO1</td><td>10</td></tr>
<tr><td></td><td></td><td>b)</td><td>Apply Backtracking for the 4 Queens problem and draw a state space tree to obtain the solution. Also, write an algorithm for the same.</td><td>CO1</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>10</td><td>a)</td><td>Find the optimal solution for the following Job Assignment problem using branch-and-bound technique.
<table>
<tr><td></td><td>J1</td><td>J2</td><td>J3</td><td>J4</td></tr>
<tr><td>P1</td><td>9</td><td>2</td><td>7</td><td>8</td></tr>
<tr><td>P2</td><td>6</td><td>4</td><td>3</td><td>7</td></tr>
<tr><td>P3</td><td>5</td><td>8</td><td>1</td><td>8</td></tr>
<tr><td>P4</td><td>7</td><td>6</td><td>9</td><td>4</td></tr>
</table></td><td>CO1</td><td>PO1</td><td>6</td></tr>
<tr><td></td><td></td><td>b)</td><td>Distinguish between P, NP, NP-complete and NP-Hard Problems. Give examples for each category.</td><td>CO1</td><td>PO1</td><td>6</td></tr>
<tr><td></td><td></td><td>c)</td><td>Use branch-and-bound technique to find optimal solution for the given Knapsack problem where capacity =10.
<table>
<tr><th>Item</th><th>Weight</th><th>Value</th></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></td><td>CO1</td><td>PO1</td><td>8</td></tr>
</table>

*****