U.S.N.										

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

Autonomous Institute Affiliated to VTU

February 2025 Semester End Main Examinations

Programme: B.E. **Semester: IV**
Branch: Computer Science and Engineering **Duration: 3 hrs.**
Course Code: 23CS4PCADA **Max Marks: 100**
Course: Analysis and Design of Algorithms

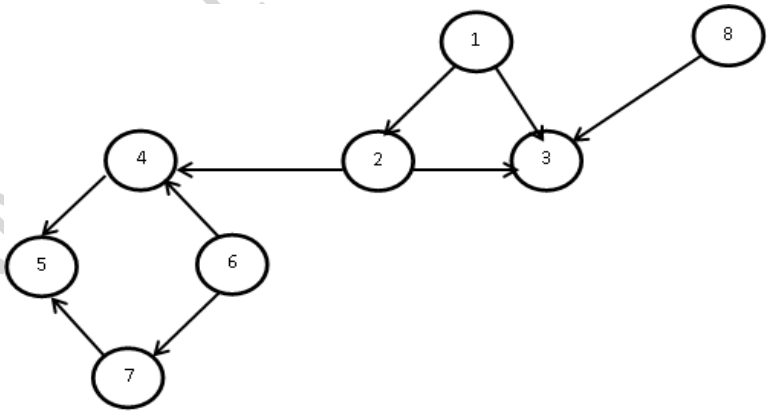
Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

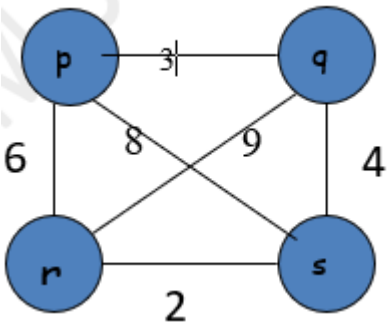
Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

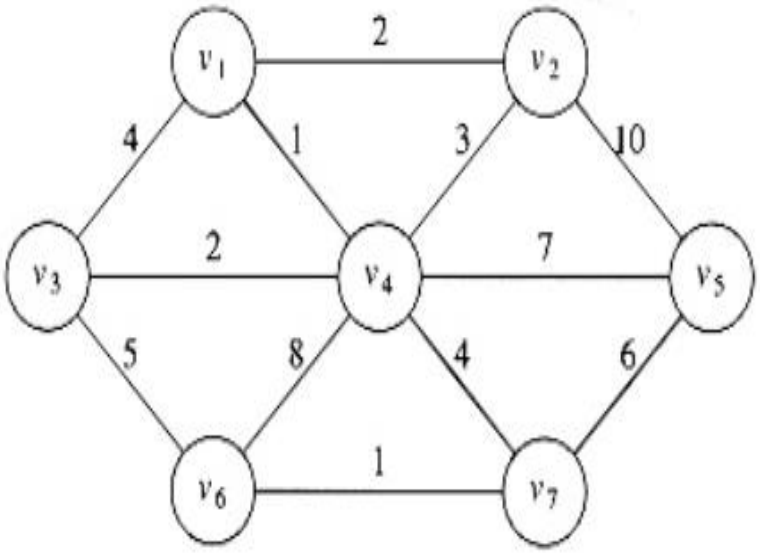
		UNIT - I	*CO*	*PO*	*Marks*
1	**a)**	By applying the steps in finding out the time complexity of non-recursive, find the time complexity for the following code: **fun(a)** **{** **int x=0;** **for(i=1;<=n;i++)** **{** **for(j=1;j<=n;j++)** **{** **if(i==j)** **{** **x=x+a[i][j];** **}** **}** **}** **printf("%d",x);** **}**	*CO1*	*PO2*	**6**
	b)	Demonstrate with an example scenario the Worst case, Best case and Average case time complexity of an algorithm.	*CO2*	*PO1*	**8**
	c)	Solve the following recurrence relation using backward substitution method: **i. x(n)= x(n/2)+n for n>1, x(1)=1** **ii. x(n)=x(n-1)+5 for n>1 and x(0)=0**	*CO2*	*PO1*	**6**
		OR			
2	**a)**	Linear Search varies its time complexity for the best case and worst case. Justify your answer.	*CO1*	*PO1*	**6**

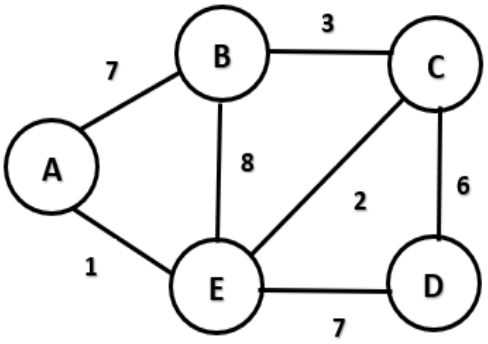
		b)	Consider the following algorithm. Procedure Secret(A[0..n−1]) // Input: An arrayA[0..n−1] of integers minval←A[0]; maxval←A[0] fori←1 to n−1do if A[i]<minval minval←A[i] if A[i]>maxval maxval←A[i] return maxval−minval i. What does this algorithm compute? ii. What is its basic operation? iii. How many times is the basic operation executed? iv. Provide an exact expression for the running time T(n) of the algorithm. v. What is the order of T(n)?	***CO2***	***PO1***	**8**

		c)	Consider the following recursive algorithm computing the sum of the first n cubes
			S(n)=1 3 +2 3 +…n 3
			Algorithm S(n)
			//Input: A positive integer n
			//Output: The sum of the first n cubes
			if n = 1 return 1
			else return S(n − 1) + n ∗ n ∗ n
			Set up and solve a recurrence relation for the number of times the algorithm's basic
			operation is executed.
				CO2	***PO1***	**6**

			UNIT - II			
	3	**a)**	Apply Decrease and Conquer technique to find Topological order for the following graph using DFS method and Source Removal method with the source vertex '1':  	***CO2***	***PO1***	**8**
		b)	Determine the number of character comparisons made by the Brute-Force pattern matching algorithm in searching for the pattern "**WOOD**" in the text "**TWO_ROADS_DIVERGED_IN_A_YELLOW_WOOD**". Also write an algorithm for the same and derive the best-case and worst-case time complexities.	***CO1***	***PO2***	**8**

		c)	Johnson Trotter is an efficient method to generate the permutations? Justify your answer with an example.	***CO2***	***PO2***	**4**
			OR			
	4	**a)**	You are given an unsorted list of distinct integers: [12, 3, 5, 7, 19, 26, 1, 8]. Use decrease and conquer algorithm to demonstrate the finding of 3rd smallest element in the list. Write the algorithm for the same	***CO2***	***PO2***	**8**
		b)	Discuss the advantages and Disadvantages of Brute force technique. Apply Brute force technique to list all tours starting from city p and find the shortest among them 	***CO2***	***PO2***	**8**
		c)	Demonstrate the multiplication of two n digit numbers using decrease by constant factor technique. Apply the same to multiply the numbers 85*18	***CO1***	***PO1***	**4**
			UNIT - III			
	5	**a)**	Is merge sort is better than quick sort in the worst case? justify your answer by deriving the time complexities for both in worst case.	***CO1***	***PO2***	**8**
		b)	Create a min heap tree for the following list of elements and sort an array: **{58, 25, 35, 38, 110, 48, 18}**	***CO2***	***PO1***	**8**
		c)	Apply Divide and Conquer technique to multiply the following two long integers: **2547 and 1605**	***CO2***	***PO1***	**4**
			OR			
	6	**a)**	For the given array, write an algorithm to determine mode using the concept of presorting and analyze its time complexity.	***CO1***	***PO2***	**6**
		b)	Construct Merge Sort tree to sort the following list of elements in the ascending order: **{10,34,22,11,54,66,33,24,25,56,77,21}**	***CO2***	***PO1***	**8**

		c)	Apply Stressen's matrix multiplication to multiply the following two matrices. $\begin{bmatrix} 5 & 6 & 1 & 2 \\ 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$ $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 7 & 9 \\ 3 & 4 & 7 & 8 \\ 1 & 3 & 5 & 7 \end{bmatrix}$	***CO2***	***PO1***	**6**
			UNIT - IV			
	7	**a)**	Several chocolates are placed in cells of an n × m board, no more than one chocolate per cell. A kid is standing at the upper left cell of the board, needs to collect as many of the chocolates as possible and bring them to the bottom right cell. On each step, the kid can move either one cell to the right or one cell down from its current location. When the kid visits a cell with a chocolate, he/she always picks up that chocolate. Find the maximum number of chocolates that the kid can collect and a path it needs to follow to do this. **1 2 3 4 5 6** **1** : , , , **Chocolate** , , **2** : , **Chocolate** , , **Chocolate** , **Chocolate** **3** : **Chocolate** , , **Chocolate** , , , **4** : , **Chocolate** , , **Chocolate** , **5** : , , , **Chocolate** , **Chocolate**	***CO2***	***PO1***	**6**
		b)	Apply Kruskal's algorithm to find minimum spanning tree for the following graph. Show the steps completely. 	***CO2***	***PO1***	**8**
		c)	Explain Spanning tree and Minimum Spanning tree with an example.	***CO2***	***PO1***	**6**
			OR			

	8	**a)**	Solve the 0/1 Knapsack problem for the following data using Dynamic programming technique. Find the objects selected for an optimal solution. Also write an algorithm for the same. **Number of objects n=5, Knapsack Capacity=6** **Item No. / Profit / Weight** **1 / 25 / 3** **2 / 20 / 2** **3 / 15 / 1** **4 / 40 / 4** **5 / 50 / 5**	***CO2***	***PO1***	**8**
		b)	Apply Dijkstra's algorithm to find shortest path from the vertex '**A**' to all other vertices for the following graph:  	***CO2***	***PO1***	**8**
		c)	Differentiate between Floyd's algorithm and Dijkstra's algorithm with an example.	***CO2***	***PO2***	**4**
			UNIT - V			
	9	**a)**	Distinguish between P, NP and NP completeness problem.	***CO3***	***PO2***	**6**
		b)	Apply Backtracking technique to find Sum of Subsets for a set **S= {12, 16, 27, 43}** and **M=55**. Represent the complete state space tree for finding all possible subsets.	***CO2***	***PO1***	**6**
		c)	Apply Branch and Bound approach to solve the Knapsack problem for the following data: **Number objects n=4, Knapsack Capacity M=10** **Item No. / Profit / Weight** **1 / 40 / 4** **2 / 42 / 7** **3 / 25 / 5** **4 / 12 / 3**	***CO2***	***PO1***	**8**
			OR			
	10	**a)**	Explain n-queens problem and its solution using backtracking with a example.	***CO2***	***PO1***	**6**
		b)	Convert below 3CNF to clique problem and also find solution for the same. Show the steps. $CNF=(x1 \vee \overline{x2} \vee \overline{x3}) \wedge (\overline{x1} \vee x2 \vee \overline{x3}) \wedge (x1 \vee x2 \vee x3)$	***CO3***	***PO2***	**6**
		c)	Describe branch and bound solution to travelling Salesman problem with example	***CO2***	***PO1***	**8**

****