U.S.N.										

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

Autonomous Institute Affiliated to VTU

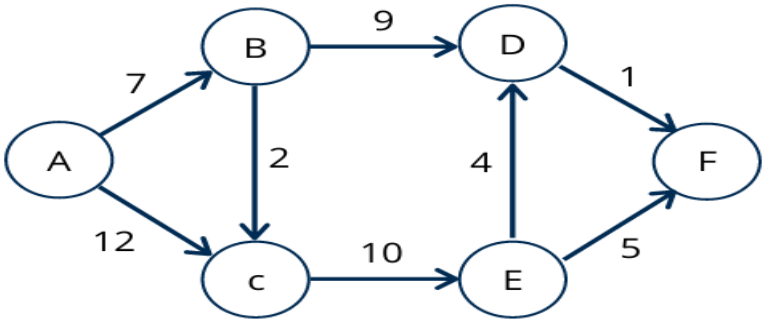
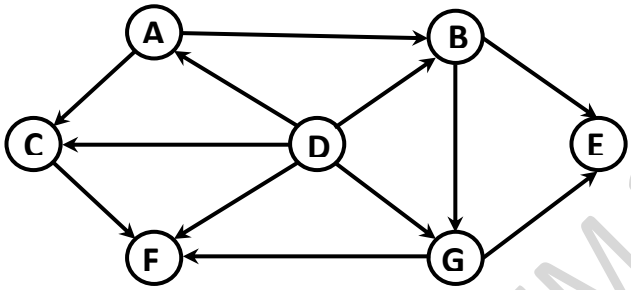
September / October 2023 Semester End Main Examinations

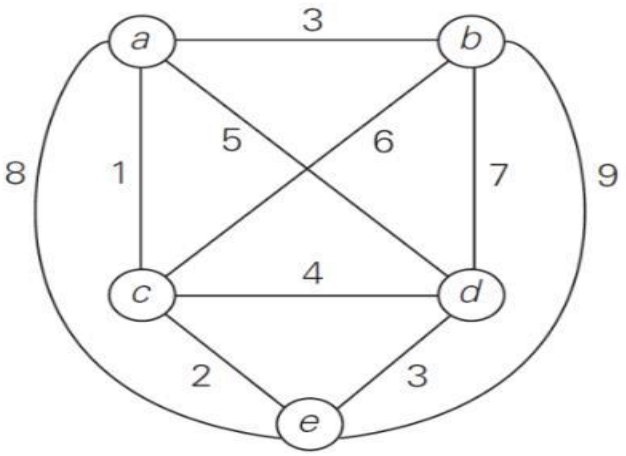
Programme: B.E. **Semester: IV**
Branch: Information Science and Engineering **Duration: 3 hrs.**
Course Code: 22IS4PCADA **Max Marks: 100**
Course: Analysis and Design of Algorithms **Date: 22.09.2023**

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)	i. Design a recursive algorithm for computing 2^n for any nonnegative integer n that is based on the formula: $2^n = 2^{n-1} + 2^{n-1}$ ii. Set up a recurrence relation for the number of additions made by the algorithm and solve it. iii. Draw a tree of recursive calls for this algorithm and count the number of calls made by the algorithm.	*CO3*	*PO3*	**08**
	b)	Discuss the general plan to find the efficiency of recursive algorithms. Apply the same to find the efficiency of solving Tower of Hanoi problem.	*CO1*	*PO1*	**08**
	c)	Apply Master's Theorem for the following i) $T(n) = 2T(n/4) + n$ ii) $T(n) = 8T(n/2) + n^3$	*CO2*	*PO2*	**04**
		UNIT - II			
2	a)	Design an algorithm used to perform partition in Quick Sort. Discuss with recurrences the efficiency of Quick Sort in Best and Worst case.	*CO3*	*PO3*	**08**
	b)	Define Spanning Tree. Discuss with time complexity the Prim's algorithm to construct Minimum Spanning Tree.	*CO1*	*PO1*	**08**
	c)	Design an algorithm to find the position of maximum element in an array using Divide and Conquer. Comment on its time complexity.	*CO3*	*PO3*	**04**
		OR			
3	a)	Write the Bubble Sort algorithm. Find the number of swaps done while sorting the following set of elements in ascending order using Bubble Sort. **6 2 11 7 5**	*CO2*	*PO2*	**08**
	b)	Explain Dijkstra's shortest path algorithm. Using the same, find the shortest path from vertex A to the remaining vertices.	*CO2*	*PO2*	**12**

						
			UNIT - III			
4	a)		Write algorithms to perform DFS and BFS traversal. Apply DFS method to find the topological order for the graph given below. 	*CO3*	*PO3*	**12**
	b)		Write the Horspool's algorithm. How many character comparisons will be made using Horspool's algorithm in searching for each of the following patterns in a text of 1000 A's? i) AAAAB ii) BAAAA	*CO2*	*PO2*	**08**
			OR			
5	a)		Write the Johnson Trotter algorithm for generating permutations. Apply the same for an input {5,6,7,8}.	*CO2*	*PO2*	**08**
	b)		With Pseudocode, discuss Horspool's String matching algorithm and analyze its time complexity.	*CO2*	*PO2*	**08**
	c)		Design an algorithm to compute the value of the smallest element in a given array using Decrease and Conquer. Comment on its time complexity.	*CO3*	*PO3*	**04**
			UNIT – IV			
6	a)		Write the algorithm for computing binomial coefficient C(n,k) using dynamic programming approach. Draw the binomial coefficient table for C(8, 3).	*CO2*	*PO2*	**08**
	b)		Design an algorithm to construct a max heap using bottom up approach. Trace by applying the same to sort the following elements **S O R T I N G**	*CO3*	*PO3*	**12**

			UNIT – V			
	7	a)	Consider the graph given below representing an instance of TSP, where source vertex is **"a"**. Solve the problem instance using Branch and Bound technique.  	*CO2*	*PO2*	**08**
		b)	Differentiate between promising and non-promising node. Construct the state-space tree using backtracking strategy to solve the following instance of the subset sum problem: A = {1,3,5,8,9} and Sum = 12	*CO3*	*PO3*	**08**
		c)	Define NP, NP-complete class of problems with examples.	*CO1*	*PO1*	**04**

****

U.S.N.										

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

Autonomous Institute Affiliated to VTU

August 2024 Semester End Main Examinations

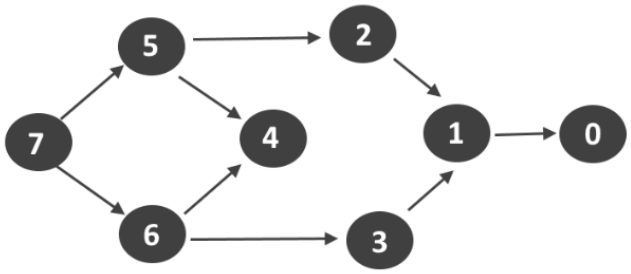
Programme: B.E. **Semester: IV**
Branch: Computer Science and Engineering **Duration: 3 hrs.**
Course Code: 23CS4PCADA **Max Marks: 100**
Course: Analysis and Design of Algorithms

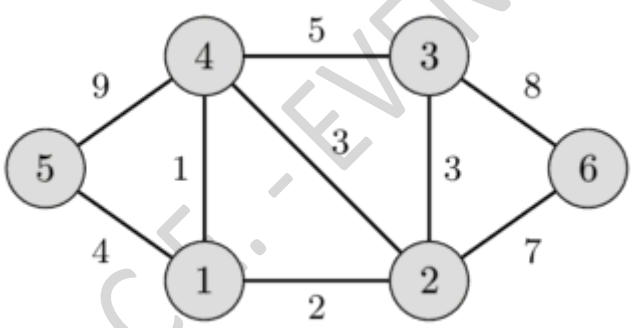
Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

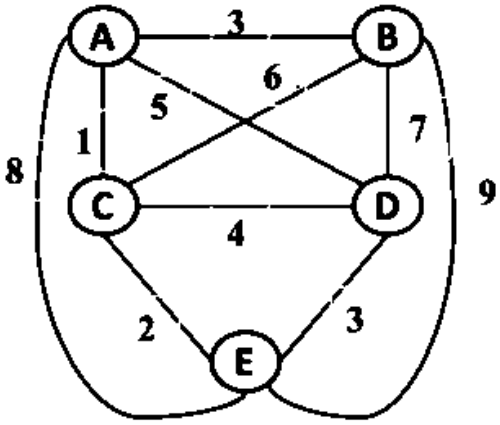
Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

		UNIT - I	CO	PO	Marks
1	a)	Derive the best case, average case and worst-case time complexities of Binary search algorithm.	CO 1	PO 2	5
	b)	Differentiate among different types of asymptotic notations with suitable example.	CO 1	PO 2	10

	c)	Find time complexity for the below algorithm:
		ALGORITHM
		UniqueElements(A[0..n − 1])
		//Determines whether all the elements in a given array are distinct
		//Input: An array A[0..n − 1]
		//Output: Returns “true” if all the elements in A are distinct
		// and “false” otherwise
		for i ←0 to n − 2 do
		for j ←i + 1 to n − 1 do
		if A[i]= A[j] return false
		return true
			CO 1	PO 2	5

		UNIT - II			
2	a)	“Johnson Trotter method is efficient method for generating permutation”? Justify the statement. Generate all permutation for the following set using Johnson Trotter algorithm: **{1,3,5,7}**	CO 2	PO 1	6
	b)	Find topological ordering of the vertices for the below directed acyclic graph using DFS method and Source removal method. 	CO 2	PO 1	8

		c)	Write an algorithm for computing the median using Decrease and Conquer technique.	*CO 2*	*PO 1*	**6**
			UNIT - III			
	3	a)	Discuss Divide and Conquer strategy. Explain Merge sort algorithm with example and give its recurrence relation.	*CO 2*	*PO 1*	**10**
		b)	Apply bottom up heap sort technique to sort the following list of elements. Also write an algorithm for the same. **{ 2, 9, 7, 6, 5, 8}**	*CO 2*	*PO 1*	**10**
			OR			
	4	a)	Strassen's matrix multiplication method is efficient compare to traditional matrix multiplication? Justify the statement with an example scenario.	*CO 1*	*PO 2*	**10**
		b)	Demonstrate all the cases of Horspool pattern matching technique. Apply the technique to search for the pattern "**ALGORITHMS**" in the text "**ANALYSIS AND DESIGN OF ALGORITHMS**". Also find its time efficiency.	*CO 1*	*PO 2*	**10**
			UNIT - IV			
	5	a)	Find the minimum spanning tree for the following graph using Kruskal's algorithm. Also write an algorithm for the same.  	*CO 2*	*PO 1*	**10**
		b)	Solve the following instance of Knapsack by Dynamic programming technique: **Number of objects n = 4** **Weights (W_i)= {2, 3, 4,5}** **Profits (P_i) = {3, 4, 5, 6}** **Knapsack Capacity C=5**	*CO 2*	*PO 1*	**10**
			OR			
	6	a)	Find the transitive closure for the relation R={(2,1),(2,3),(3,1),(3,4),(4,1),(4,3) on set A={1,2,3.4} using Warshall's algorithm. Also write an algorithm for the same and find its time complexity.	*CO 2*	*PO 1*	**10**

		b)	Consider the five-symbol alphabet {A, B, C, D, _ } with the following occurrence frequencies in a text made up of these symbols. **Symbol** | **A** | **B** | **C** | **D** | **_** **Frequency** | **0.35** | **0.1** | **0.2** | **0.2** | **0.15** Construct Huffman tree and generate the code for the symbols. Also write Huffman's algorithm.	***CO 2***	***PO 1***	**10**
			UNIT - V			
	7	a)	Solve the travelling salesman problem for the following graph using Branch and Bound technique.  	***CO 2***	***PO 1***	**10**
		b)	Apply Backtracking technique to find the sum of subsets for the set s={3,5,6,7} and d=15. Draw a state space tree.	***CO 2***	***PO 1***	**10**

******

U.S.N.										

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

Autonomous Institute Affiliated to VTU

August 2024 Semester End Main Examinations

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

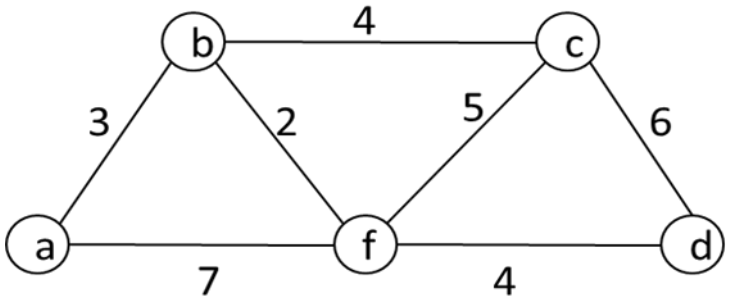
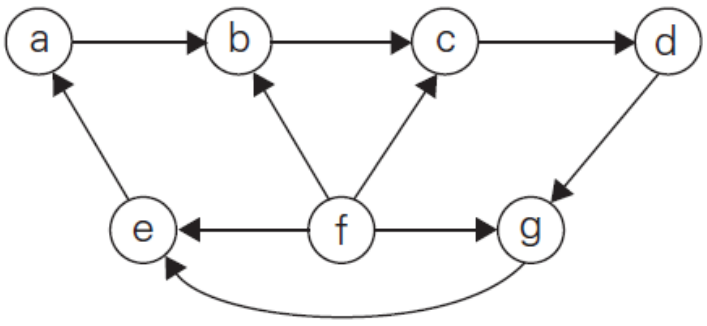
Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

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)	Explicate a general plan for analyzing the time efficiency of Nonrecursive algorithms. List the following functions according to their order of growth from the lowest to the highest: $(n-2)!$, $5 \lg(n+100)^{10}$, $0.001n^4 + 3n^3 + 1$, $\ln^2 n$, 3^n	*CO2*	*PO2*	**10**

	b)	Consider the algorithm given below and answer the questions that follow:
		ALGORITHM Enigma(A[0..n − 1, 0..n − 1])
		//Input: A matrix A[0..n − 1, 0..n − 1] of real numbers
		for i ←0 to n − 2 do
		for j ←i + 1 to n − 1 do
		if A[i, j] != A[j, i]
		return false
		return true
		i) What does this algorithm compute?
		ii) What is its basic operation?
		iii) How many times is the basic operation executed?
		iv) What is the efficiency class of this algorithm?
		v) Suggest an improvement, or a better algorithm altogether, and indicate its efficiency class. If you cannot do it, try to prove that, in fact, it cannot be done.
			CO2	*PO2*	**10**

		UNIT - II			
2	a)	Write the Selection Sort algorithm. Sort the list {E, X, A, M, P, L, E} in alphabetical order by selection sort.	*CO2*	*PO2*	**10**
	b)	Write Dijikstra's algorithm to find all the shortest paths from a given source vertex to all other vertices in a graph. Trace the following graph to get shortest path from vertex 'a' to all other vertices.	*CO3*	*PO3*	**10**

						
			OR			
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)	Write the brute force string matching algorithm. Determine the number of character comparisons made by the brute-force string matching algorithm in searching for the pattern GANDHI in the text {THERE IS MORE TO LIFE THAN INCREASING ITS SPEED}. Assume that the length of the text is 47 characters long and is known before the search starts.	*CO2*	*PO2*	**10**
		UNIT - III			
4	a)	Apply Horspool's algorithm to search for the pattern BARBER in some text considering the following cases. i) No match at all ii) Match found but not with last character iii) Only last (1 or more) character is matching iv) Match found with last as well as with other characters	*CO3*	*PO3*	**12**
	b)	With suitable example, explain the depth first search and breadth first search algorithms.	*CO1*	*PO1*	**08**
		OR			
5	a)	Write an algorithm for breadth first search. Apply the DFS-based algorithm to solve the topological sorting problem for the following digraph. 	*CO2*	*PO2*	**08**
	b)	For the input {30, 20, 56, 75, 31, 19} and hash function h(K) = K mod 11 i) construct the open hash table. ii) find the largest number of key comparisons in a successful search in this table.	*CO2*	*PO2*	**08**

			iii) find the average number of key comparisons in a successful search in this table.			
		c)	Generate all permutations of the given set {3, 5, 7} using Johnson-Trotter algorithm.	*CO2*	*PO2*	**04**
			UNIT - IV			
	6	a)	Solve the all-pairs shortest-path problem for the digraph with the following weight matrix: $\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 2 & \infty & 1 & 8 \\ 6 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & \infty & \infty & \infty & 0 \end{pmatrix} \end{array}$	*CO2*	*PO2*	**10**
		b)	Write the two-stage heapsort algorithm. Apply the same to sort the list {2, 9, 7, 6, 5, 8}.	*CO3*	*PO3*	**10**
			UNIT - V			
	7	a)	Write the decision tree for three-element selection sort.	*CO1*	*PO1*	**06**
		b)	Distinguish between P, NP and NP-Complete problems. Give example for each category.	*CO1*	*PO1*	**06**
		c)	Solve the following job assignment problem using branch and bound design technique. $\begin{array}{cccc} \text{job 1} & \text{job 2} & \text{job 3} & \text{job 4} \end{array}$ $\begin{bmatrix} 9 & 2 & 7 & 8 \\ 6 & 4 & 3 & 7 \\ 5 & 8 & 1 & 8 \\ 7 & 6 & 9 & 4 \end{bmatrix}$ person *a* person *b* person *c* person *d*	*CO2*	*PO2*	**08**

****

U.S.N.										

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

Autonomous Institute Affiliated to VTU

September / October 2023 Semester End Main Examinations

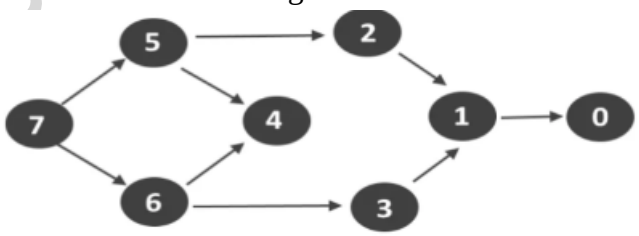
Programme: B.E. **Semester: IV**
Branch: Computer Science & Engineering **Duration: 3 hrs.**
Course Code: 22CS4PCADA **Max Marks: 100**
Course: Analysis and Design of Algorithms **Date: 22.09.2023**

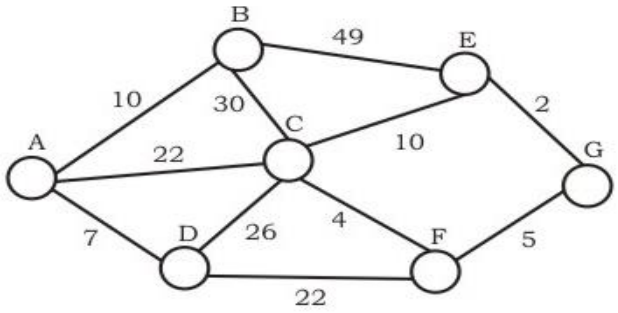
Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

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.

			CO	PO	Marks
		UNIT - I			
1	a)	Describe Asymptotic notations with an example, equation and graph.	CO1	PO1	**6**
	b)	Design a recursive algorithm to find maximum and minimum element in an array and derive its time complexity.	CO1	PO1,2,3	**8**

	c)	Analyze the code given below and find the time complexity:
		`void function (int n)`
		`{`
		`  int count = 0;`
		`  for (int i=n/2; i<=n; i++)`
		`  for (int j=1; j<=n; j = 2 * j)`
		`  for (int k=1; k<=n; k = k * 2)`
		`  count++;`
		`}`
			CO1	PO2	**6**

		UNIT - II			
2	a)	Mention the applications of Depth First Search (DFS) algorithm. Apply DFS method to find the topological sequence for the graph shown below. Also write an algorithm for the same. 	CO2,4	PO1,2,3	**10**
	b)	Solve the following assignment problem using Exhaustive Search technique, whose matrix for assigning four jobs to four persons are given: | | J1 | J2 | J3 | J4 | | --- | --- | --- | --- | --- | | Person A | 3 | 21 | 7 | 9 | | Person B | 18 | 13 | 10 | 15 | | Person C | 29 | 19 | 17 | 12 | | Person D | 32 | 30 | 26 | 28 |	CO2,4	PO1,2,3	**10**

			OR			
	3	a)	Demonstrate with an example Brute Force pattern matching technique. Write an algorithm for the same and find its time complexity.	*CO2,4*	*PO1,2,3*	**10**
		b)	Write a program to check whether a particular given node is reachable from a given source node using DFS traversal technique. If yes, print "Node is accessible", otherwise print "Node is not accessible".	***CO2***	***PO1***	**10**
			UNIT - III			
	4	a)	Apply Merge sort to sort the list {E, X, A, M, P, L, E} in alphabetical order.	*CO2,4*	*PO1,2,3*	**10**
		b)	Construct a Max Heap for the following list of keys and sort the list using Heap Sort technique. Write the algorithm for Max Heap. **{2,9,7,6,5,8}**	*CO2,4*	*PO1,2,3*	**10**
			OR			
	5	a)	State Horner's rule with an algorithm. Apply Horner's rule to evaluate the following polynomial: $5x^4 + 2x^3 - 3x^2 + x - 7$ at the point x=3.	*CO2*	*PO1,2*	**10**
		b)	Demonstrate the Strassen's Matrix Multiplication method with an example.	*CO2*	*PO2*	**10**
			UNIT - IV			
	6	a)	Analyze time efficiency of Prim's algorithm. Apply Prim's algorithm to find the minimum cost spanning tree for the graph shown below: 	*CO2,4*	*PO1,2,3*	**10**
		b)	Construct a Huffman tree and find the code word for the following data: <table><tr><td>Character</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>-</td></tr><tr><td>Frequency</td><td>0.5</td><td>0.35</td><td>0.5</td><td>0.1</td><td>0.4</td><td>0.2</td></tr></table> Using above code, Encode the text DAD_CBE and decode the text 1000010111001010.	*CO2,4*	*PO1,2*	**10**
			UNIT - V			
	7	a)	Solve the following instance of 0/1 Knapsack problem using Branch and Bound with capacity C=10. Items={1,2,3,4} Weights={4,7,5,3} Values={$40, $42, $25,$12}	*CO2,4*	*PO1,2*	**10**
		b)	Describe the P and NP class problems with examples. Illustrate NP Completeness proof by Reduction.	*CO3*	*PO1,2*	**10**

****

U.S.N.										

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

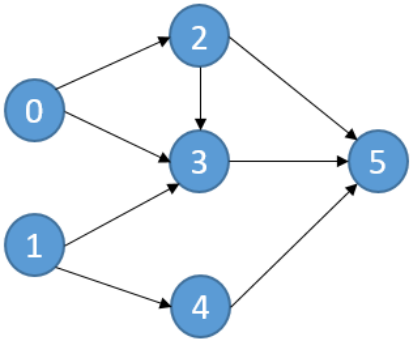
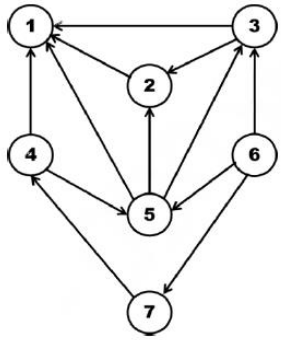
Autonomous Institute Affiliated to VTU

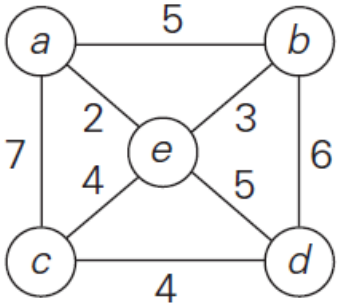
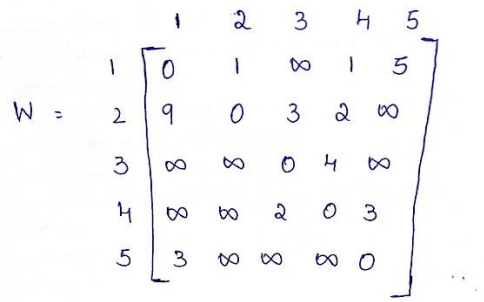
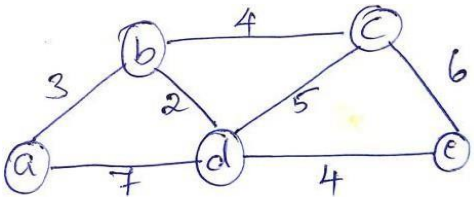
August 2024 Semester End Main Examinations

Programme: B.E. **Semester: IV**
Branch: Information Science and Engineering **Duration: 3 hrs.**
Course Code: 23IS4PCADA **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)	Explain different asymptotic notations used to represent the time complexities with suitable examples.	CO1	-	10
	b)	Outline selection sort and Bubble sort algorithms with example.	CO2	PO1	10
		UNIT - II			
2	a)	Write the quick sort algorithm. Apply the same to sort the list {E, X, A, M, P, L, E} in alphabetical order.	CO2	PO1	12
	b)	Apply source removal method to solve the following topological sorting problem. 	CO2	PO1	08
		OR			
3	a)	Find the BFS and DFS traversals starting from vertex 6 for the following Graph. Also, write the BFS and DFS Algorithms. 	CO2	PO1	12

		b)	Apply Merge sort algorithm to sort the numbers {14, 91, 07, 01, 10, 29, 08, 02}. Show the Merge call tree for the same.	*CO2*	*PO1*	**08**
			UNIT - III			
	4	a)	Write the Prim's algorithm. Apply Prim's algorithm to the following graph. Start from vertex 'a'. 	*CO3*	*PO1*	**10**
		b)	Solve the following instance of Knapsack problem using dynamic programming. Knapsack Capacity M=10 Item 1 2 3 4 Weight 4 7 5 3 Profit 40 42 25 12	*CO2*	*PO1*	**10**
			OR			
	5	a)	Apply Floyd's algorithm to find all pairs shortest path for the given adjacency matrix. 	*CO2*	*PO1*	**10**
		b)	Using Dijkstra's algorithm, trace the following graph to get shortest path from vertex 'a' to all other vertices. Also, write the algorithm. 	*CO3*	*PO1*	**10**
			UNIT - IV			
	6	a)	Write the Heap Sort Algorithm. Show how the following numbers are sorted using Heap Sort {11, 44, 10, 65, 50, 6, 88, 3}.	*CO3*	*PO1*	**12**

<table>
<tr><td></td><td></td><td>b)</td><td>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</td><td>CO2</td><td>PO1</td><td>08</td></tr>
<tr><td></td><td></td><td></td><td>UNIT - V</td><td></td><td></td><td></td></tr>
<tr><td></td><td>7</td><td>a)</td><td>Differentiate between NP Hard and NP Complete Problems.</td><td>CO2</td><td>PO2</td><td>06</td></tr>
<tr><td></td><td></td><td>b)</td><td>Find any one solution to 4-queens problem using backtracking. Draw the state-space tree.</td><td>CO3</td><td>PO1</td><td>05</td></tr>
<tr><td></td><td></td><td>c)</td><td>Obtain the optimal solution for the given job assignment problem using Branch and Bound method.
<table>
<tr><td></td><td>JOB1</td><td>JOB2</td><td>JOB3</td><td>JOB4</td></tr>
<tr><td>Person A</td><td>9</td><td>2</td><td>7</td><td>8</td></tr>
<tr><td>Person B</td><td>6</td><td>4</td><td>3</td><td>7</td></tr>
<tr><td>Person C</td><td>5</td><td>8</td><td>1</td><td>8</td></tr>
<tr><td>Person D</td><td>7</td><td>6</td><td>9</td><td>4</td></tr>
</table></td><td>CO2</td><td>PO1</td><td>09</td></tr>
</table>

****

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: 22CS4PCADA **Max Marks: 100**
Course: Analysis and Design of Algorithms

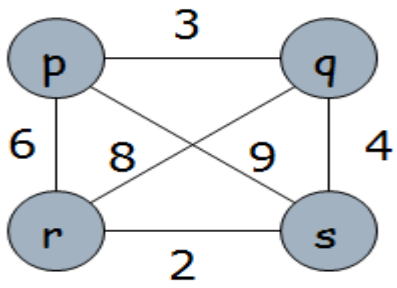
Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

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)		Demonstrate worst case best case and average case scenario complexity of an algorithm with an example.	CO1	PO 1	**8**
	b)		With the help of a flowchart, explain the various stages of algorithm design process.	CO1	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	CO1	PO1	**4**
			OR			

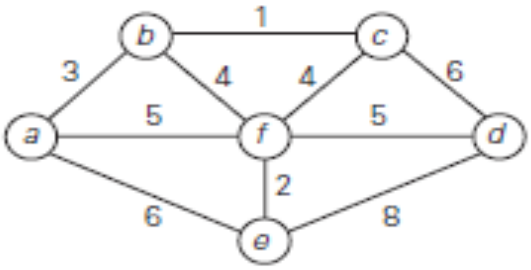
2	a)		By applying the steps in finding out the time complexity of non-recursive algorithm shown below.Find the time complexity for the following code.
			i. void main()
			{ int i, j, k = 0;
			for (i = n/2; i <= n; i++)
			{
			for (j = 2; j <= n; j = j * 2)
			{
			k = k + n / 2;
			}
			}
			}
				CO2	PO1	**4**

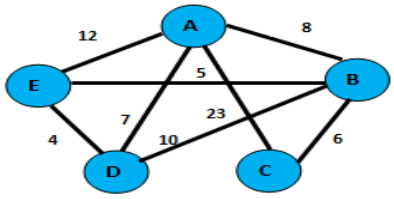
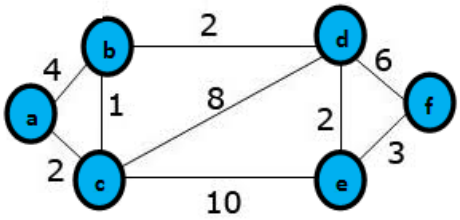
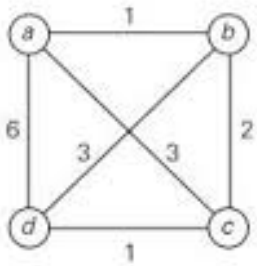
	b)		Write recursive algorithm for Towers of Hanoi problem for "n" disks. Draw the recursion tree for n=3 and showing the order of moves	CO2	PO1	**6**
	c)		Apply selection sort technique to sort the list {O, N, L, I, N, E, T, E, S, T} in alphabetical order showing the output of each pass during the sorting process. Write an algorithm for the same and find its time complexity	CO2	PO1	**10**

			UNIT - II			
	3	a)	Apply exhaustive search technique to list all tours starting from city 'p' and find the minimum cost route among them.  	*CO2*	*PO1*	**4**
		b)	Differentiate between different variations of Decrease and Conquer technique with an example	*CO2*	*PO2*	**6**
		c)	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*	**10**
			OR			
	4	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*	**10**
		b)	Apply Johnson Trotter method to generate permutations for the following set. 1,2,3,4	*CO2*	*PO1*	**6**
		c)	Apply Exhaustive Search technique to solve the following instance of Knapsack problem: Number of objects N=4, weights of four objects= {7, 3, 4, 5) and profits= {42, 12, 40, 25} with the capacity of Knapsack W=10	*CO2*	*PO1*	**4**
			UNIT - III			
	5	a)	Given the numbers {10,34,22,11,54,66,33,24,25,56,77,21}. Construct MergeSort tree to sort these numbers in the ascending order. Also write an algorithm for Merge sort.	*CO2*	*PO2*	**8**
		b)	For the given array, write an algorithm to determine mode using the concept of presorting and analyze its time complexity	*CO1*	*PO1*	**6**
		c)	Briefly explain different variations of Transform and Conquer technique, explain each with an example.	*CO2*	*PO1*	**6**
			OR			
	6	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. Also, write the algorithm for the same. {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**
		UNIT - IV			
7	a)	Suppose the knapsack problem is solved by Dynamic programing technique and the solution table is given below. Explain step by step process of selecting objects to get optimal solution. Consider number of objects=4, Weight={1,5,3,4} for the items with the number (1,2,3,4), Capacity of Knapsack=8	*CO2*	*PO1*	**6**
	b)	Apply Krushkals's algorithm to find minimum spanning tree for the following graph. Also write an algorithm for the same	*CO2*	*PO1*	**8**
	c)	Suppose the string below is to be sent over a network. Construct a Huffman tree and find the code word for each character. Justify how Huffman tree reduces the string size through encoding compare to sending original string. BCAADDDCCACACAC	*CO2*	*PO1*	**6**
		OR			
8	a)	Design Dynamic programming based algorithm to find all pair shortest paths. Apply the same to the below graph	*CO2*	*PO1*	**10**

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	15	15	15	15	15	15	15	15
2	0	15	15	15	15	15	25	25	25
3	0	15	15	15	24	24	25	25	25
4	0	15	15	15	24	24	25	25	29



						
		b)	Design a Greedy algorithm for finding single source shortest paths. Apply the same on the below graph to find shortest paths from vertex 'A' to all other nodes. 	*CO2*	*PO1*	**10**
			UNIT - V			
	9	a)	Write the state space tree for finding sum of subset for the set X={5,8,13} with d=13 using Backtracking technique.	*CO2*	*PO1*	**6**
		b)	Distinguish between P, NP and NP completeness problem	*CO3*	*PO2*	**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)	Apply backtracking approach to write state space tree to find sum of subsets for set S={5, 5, 10} and d=10.	*CO2*	*PO1*	**10**
		b)	Apply branch and bound technique for the travelling salesman problem to the below graph. 	*CO2*	*PO1*	**10**

******

U.S.N.										

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

Autonomous Institute Affiliated to VTU

February 2025 Semester End Main Examinations

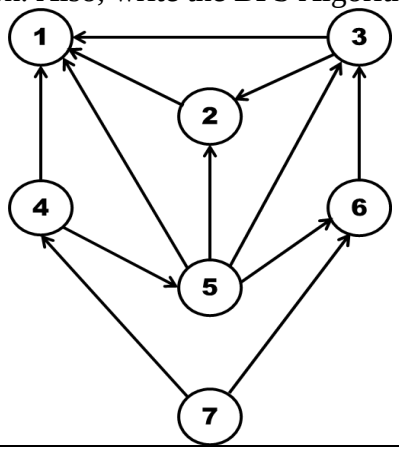
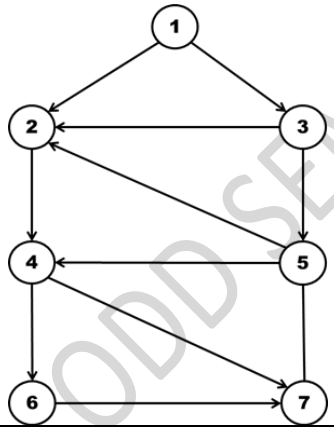
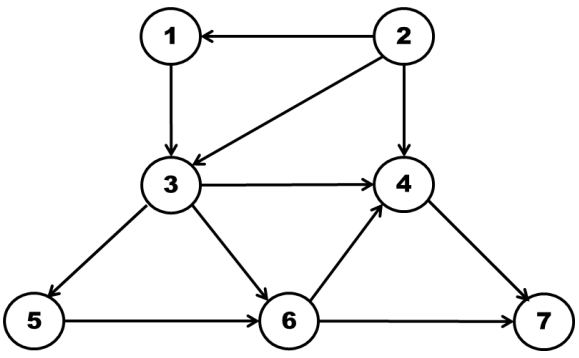
Programme: B.E. **Semester: IV**
Branch: Information Science and Engineering **Duration: 3 hrs.**
Course Code: 23IS4PCADA **Max Marks: 100**
Course: Analysis and Design of Algorithms

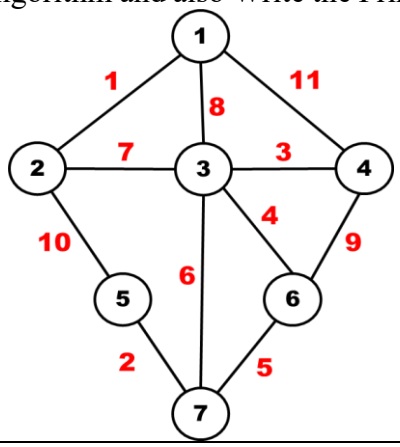
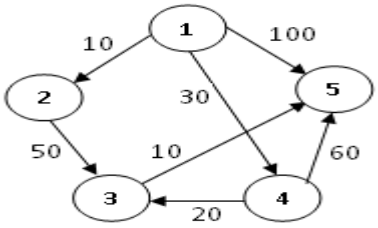
Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

		UNIT - 1	CO	PO	Marks
1	a)	Discuss the Asymptotic Notations with their definitions.	*CO2*	*PO1*	**6**
	b)	Find the Time Complexity for the following Algorithms: (i) Factorial of a given number (ii) Tower of Hanoi	*CO2*	*PO1*	**8**
	c)	Compare Bubble Sort and Selection Sort in the Best and Worst cases.	*CO2*	*PO2*	**6**
		OR			
2	a)	Design a recursive algorithm for computing 2^n for a non-negative integer n, based on the formula $2^n = 2^{n-1} + 2^{n-1}$. Setup a recurrence relation for the number of additions made by the algorithm and solve it.	*CO3*	*PO3*	**8**

	b)	Find algorithm efficiency with respect to time for the following algorithms:
		1.
		Algo_X(n)
		{
		x=0;
		for(i=0; x<=n; i++)
		x= x+i;
		}
		2.
		Algo_Hanoi (disk, source, dest, aux)
		{
		IF disk == 1, THEN
		move disk from source to dest
		ELSE
		Hanoi (disk - 1, source, aux, dest)
		move disk from source to dest
		Hanoi (disk - 1, aux, dest, source)
		END IF
		}
			CO2	*PO1*	**8**

	c)	Use recursion tree to solve T(n) = 2T(n/2) + cn, where c > 0 is constant.	*CO2*	*PO1*	**4**
		UNIT - 2			
3	a)	Show how the following numbers gets sorted using Merge Sort:	*CO1*	*PO2*	**8**

			72 46 24 57 12 68 07 18 Also, write the Merge sort Algorithm.			
		b)	Find the BFS traversals starting from vertex 7 for the following Graph. Also, write the BFS Algorithm. 	*CO3*	*PO3*	**7**
		c)	Find the Topological Sequence for the following Graph [using any method]. 	*CO3*	*PO3*	**5**
			OR			
	4	a)	Show how the following numbers gets sorted using Quick Sort: 84 23 68 09 96 66 05 25 Also, write the Quick Sort Algorithm.	*CO1*	*PO1*	**12**
		b)	Find the DFS traversal starting from vertex 2 for the following Graph. Also, write the DFS Algorithm. 	*CO3*	*PO3*	**8**

			UNIT - 3			
	5	a)	Find the Minimum Spanning Tree for the following Graph using Prim's Algorithm and also Write the Prim's Algorithm.  	*CO3*	*PO3*	**10**
		b)	Solve the following 0/1 Knapsack problem using dynamic programming: **P= (11, 7, 9, 14) W= (1, 5, 4, 6) C=10 n=4**	*CO2*	*PO1*	**10**
			OR			
	6	a)	Apply Dijkstra's algorithm to find shortest path from the vertex 1 to all other vertices for the following graph:  	*CO3*	*PO3*	**6**
		b)	Construct a Huffman tree and find the Huffman code for each character shown below: **Character / Frequency** **A 10** **B 15** **C 12** **D 3** **E 4** **F 13** **G 1**	*CO1*	*PO1*	**7**
		c)	Apply Floyd's Algorithm to find the All-Pair Shortest Path for the following: **1: 0, 5, ∞, ∞** **2: 50, 0, 15, 5** **3: 30, ∞, 0, 15** **4: 15, ∞, 5, 0**	*CO3*	*PO3*	**7**

<table>
<tr><td></td><td></td><td></td><td>UNIT - 4</td><td></td><td></td><td></td></tr>
<tr><td></td><td>7</td><td>a)</td><td>Apply Boyer Moore algorithm to search the given substring in the main string. Also, write the number of shifts required during searching.
Main String = “MISS MISS IN MISSISSIPPI”
Substring = “MISSI”</td><td>CO3</td><td>PO3</td><td>8</td></tr>
<tr><td></td><td></td><td>b)</td><td>Show how the following numbers are sorted by Heap Sort.
23 74 06 68 12 66 10 16</td><td>CO1</td><td>PO1</td><td>6</td></tr>
<tr><td></td><td></td><td>c)</td><td>Construct a Hash Table by Linear Probing/Closed Hashing for the following words. Consider the size and the hash table as 10.
WHERE, IS, NOW, THIS, AN, THAT, HOW, AND</td><td>CO1</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>8</td><td>a)</td><td>Construct an AVL tree for the list: {6, 5, 4, 3, 2, 1} by inserting their elements successively, starting with an empty tree.</td><td>CO2</td><td>PO1</td><td>6</td></tr>
<tr><td></td><td></td><td>b)</td><td>Differentiate between open hashing and separate chaining.</td><td>CO1</td><td>PO2</td><td>6</td></tr>
<tr><td></td><td></td><td>c)</td><td>Given the input {30, 20, 56, 75, 31, 19} and hash function h(K) = K mod 11, answer the following questions:
i) Construct the open hash table.
ii) Find the largest number of key comparisons in a successful search in this table.
iii) Find the average number of key comparisons in a successful search in this table.</td><td>CO1</td><td>PO1</td><td>8</td></tr>
<tr><td></td><td></td><td></td><td>UNIT - 5</td><td></td><td></td><td></td></tr>
<tr><td></td><td>9</td><td>a)</td><td>Show the solution for 4-Queens problem using Backtracking and write an algorithm for n-Queens problem using Backtracking.</td><td>CO3</td><td>PO3</td><td>8</td></tr>
<tr><td></td><td></td><td>b)</td><td>Solve the following Job Assignment Problem using the branch-and-bound technique:
<table><tr><th></th><th>Job 1</th><th>Job 2</th><th>Job 3</th><th>Job 4</th></tr><tr><td>Person 1</td><td>5</td><td>6</td><td>9</td><td>7</td></tr><tr><td>Person 2</td><td>8</td><td>4</td><td>2</td><td>6</td></tr><tr><td>Person 3</td><td>1</td><td>3</td><td>7</td><td>9</td></tr><tr><td>Person 4</td><td>9</td><td>6</td><td>7</td><td>4</td></tr></table></td><td>CO3</td><td>PO3</td><td>7</td></tr>
<tr><td></td><td></td><td>c)</td><td>Discuss the concept of P, NP, NP-Complete and NP-Hard Problems.</td><td>CO2</td><td>PO2</td><td>5</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>Apply backtracking to solve the following instance of the subset-sum problem S = {3, 5, 6, 7} and d =15.</td><td>CO2</td><td>PO1</td><td>8</td></tr>
<tr><td></td><td></td><td>b)</td><td>With the help of a state space tree, solve the following instance of Knapsack problem by the branch and bound algorithm. Knapsack Capacity W = 10
<table><tr><th>Item No.</th><th>1</th><th>2</th><th>3</th><th>4</th></tr><tr><td>Weight</td><td>4</td><td>7</td><td>5</td><td>3</td></tr><tr><td>Value</td><td>40</td><td>42</td><td>25</td><td>12</td></tr></table></td><td>CO1</td><td>PO1</td><td>8</td></tr>
<tr><td></td><td></td><td>c)</td><td>Distinguish between P, NP and NP-Complete problems. Give example for each category.</td><td>CO1</td><td>PO2</td><td>4</td></tr>
</table>

******

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

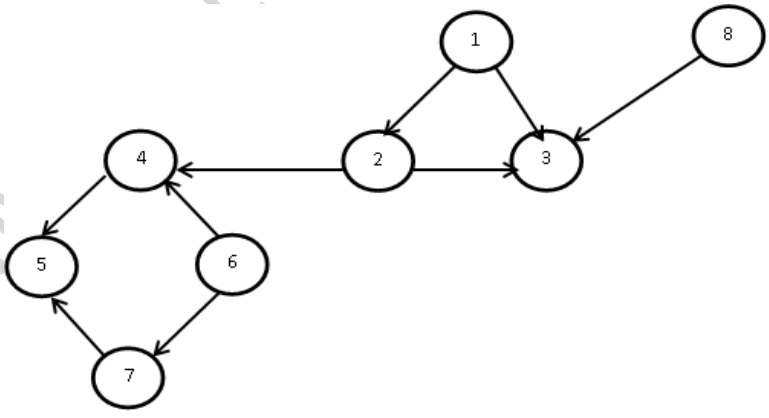
Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

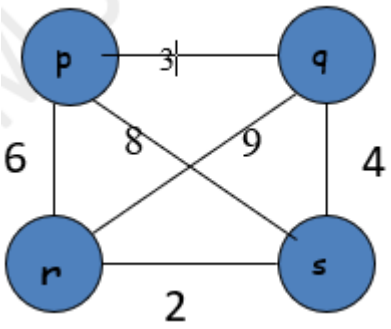
Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

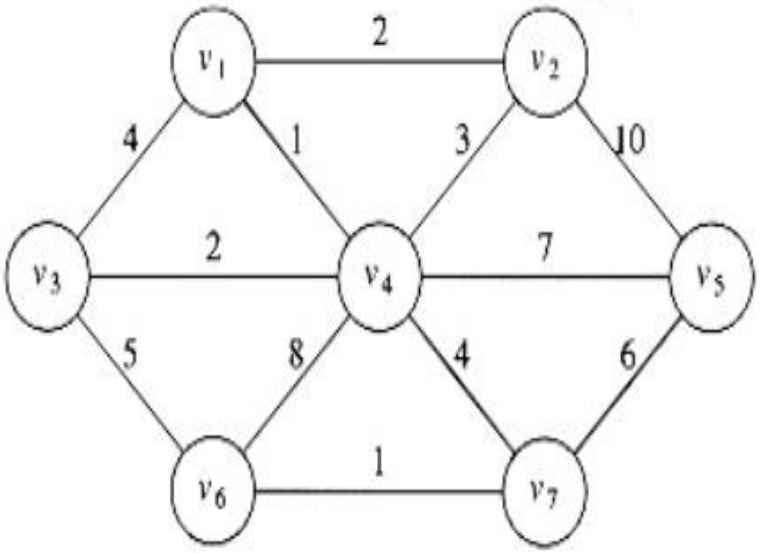
		UNIT - I	*CO*	*PO*	*Marks*
1	**a)**	By applying the steps in finding out the time complexity of non-recursive, find the time complexity for the following code: **fun(a)** **{** **int x=0;** **for(i=1;<=n;i++)** **{** **for(j=1;j<=n;j++)** **{** **if(i==j)** **{** **x=x+a[i][j];** **}** **}** **}** **printf("%d",x);** **}**	*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**

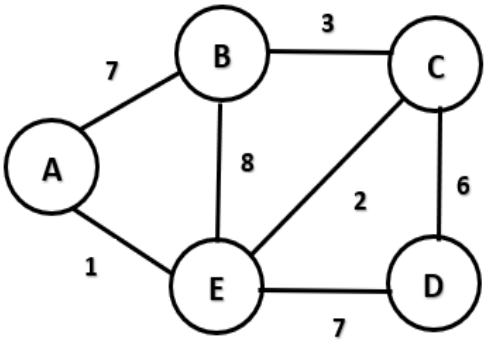
		b)	Consider the following algorithm. Procedure Secret(A[0..n−1]) // Input: An arrayA[0..n−1] of integers minval←A[0]; maxval←A[0] fori←1 to n−1do if A[i]<minval minval←A[i] if A[i]>maxval maxval←A[i] return maxval−minval i. What does this algorithm compute? ii. What is its basic operation? iii. How many 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**

****