Satyug Darshan Institute of Technology,Faridabad

Programme: B.Tech (CE) **Year:/Semester: 2nd/ 4th**

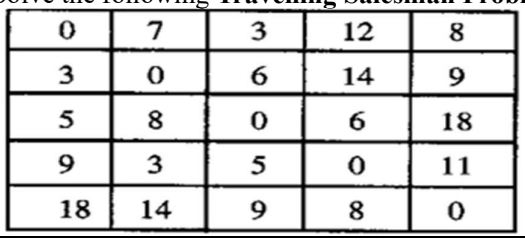
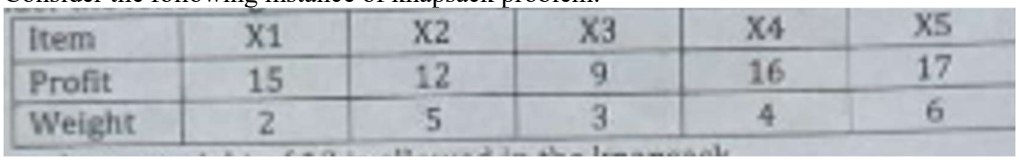
Subject Name : Design and Analysis of Algorithm **Subject Code: CEU-208-V**

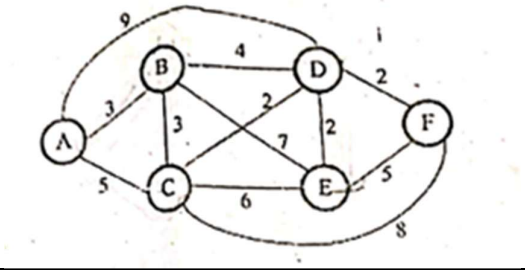
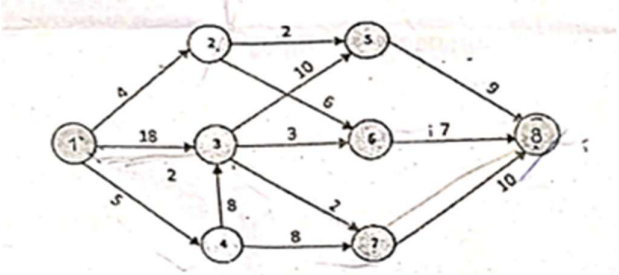
	Unit-1	**Blooms Texonomy**
	Short Question	
S. No	**Questions**	
1	Write the recurrence relation for **ternary search** and also solve it.	KL1
2	Classify big **'O'** and **small 'o'** asymptotic notations.	KL2
3	Discuss how to perform performance analysis of an algorithm.	KL2
4	Describe time and space trade off.	KL2
5	Let $f(n)$and $g(n)$be asymptotically positive functions. Prove or disprove: $f(n) = O(g(n))$ implies $g(n) = O(f(n))$.	KL2

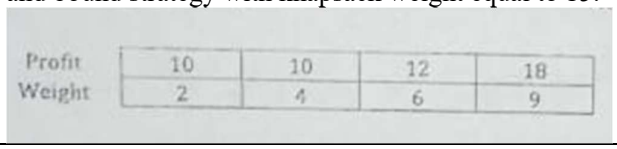
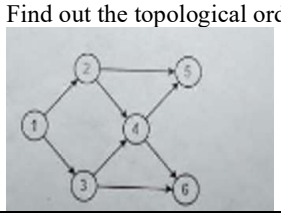
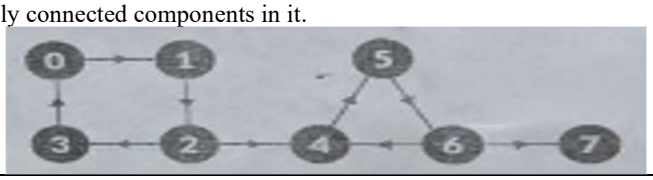
6	Write the time complexity of the following code? Justify your answer.
	int i, j, k = 0;
	
	for (i = n / 2; i <= n; i++)
	{
	for (j = 2; j <= n; j = j * 2)
	{
	k = k + n / 2;
	}
	}
		KL2

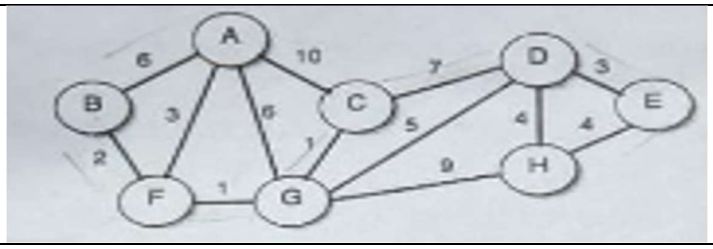
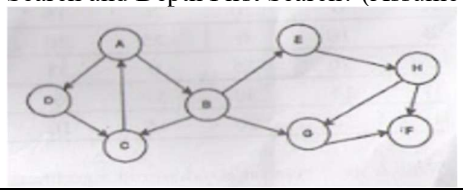
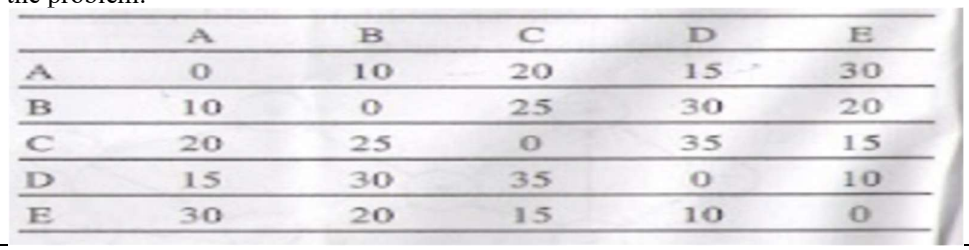
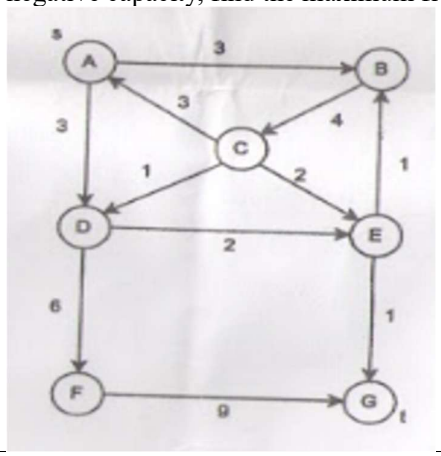
7	Sort the following functions in the decreasing order of their asymptotic (Big-O) complexity: • $f_1(n) = n^{\sqrt{n}}$ • $f_2(n) = 2^n$ • $f_3(n) = (1.000001)^n$ • $f_4(n) = n^{10} \cdot 2^{(n/2)}$	KL2
8	Write the recurrence relation for **n-array search** and why is it not preferred?	KL2
9	Write the following functions in the **increasing order of asymptotic complexity**: • $f_1(n) = 2^n$ • $f_2(n) = n^{3/2}$ • $f_3(n) = n\log\ n$ • $f_4(n) = n^{\log n}$	KL2
10	Explain the significance of Asymptotic notation.	KL2
11	Arrange the following complexity in **increasing order**: n, log n, n^3, n^2 log n, 1/log n, 2^n	KL2
12	Explain Dominance rule with example	KL2
13	Differentiate between O-notation and Omega-notation with the help of an appropriate example.	KL2
14	Describe the properties of an algorithm with an example.	KL2
15	List out the steps that need to design an algorithm.	KL1
	Long Questions	
1	Solve the following recurrence relations: (i) T(n) = T(n − 1) + n (ii) T(n) = T(n / 2) + 1 (iii) T(n) = 3T(n / 9) + n^3 (iv) T(n) = T(n / 3) + T(2n / 3) + n	KL3
2	Solve: $T(n) = 2T(n/2) + n\log\ n$	KL3

3	The running time of an algorithm P1 and P2 are represented by the following recurrence relations: **P1:** If $n \leq 3$then $T(n) = n$, else $T(n) = T(n/3) + cn$ **P2:** If $n \leq 1$then $T(n) = 1$, else $T(n) = T(n-1) + \log\ n$ Compute the time complexity of both the algorithms **P1** and **P2**.	KL3
4	Can the master method be applied to the recurrence $T(n) = 4T(n/2) + n^2 \log\ n$ Why or why not? Give an asymptotic upper bound for this recurrence.	KL3
5	**Solve the following recurrence relation:** (i) T(n) = 2T(n/2) + log n, with T(1) = 1 (ii) T(n) = 2T(n/2) + 1 if n > 2 and T(n) = 2 if 0 < n ≤ 2	KL3
6	How are time and space trade-offs used to optimize the performance of an algorithm? Provide an example of an algorithm that optimizes time at the expense of space and vice-versa.	
7	Explain **dominance rule (Merge–Purge)** with an appropriate example.	KL2
8	**Solve the following recurrence relations:** **(i)** $T(n) = T(\sqrt{n}) + n$ **(ii)** $T(n) = 7T(\lfloor n/3 \rfloor) + n^2$	KL3
9	Explain master method for computing complexity of recurrence relations. Solve the following recurrences: (i) T(n) = 3T(n/4) + n log n (ii) T(n) = T(√n) + n	KL3
10	Solve: $T(n) = 2T(n/2) + n/\log\ n$	KL3
11	Solve: $T(n) = T(n/2) + T(n/4) + n$	KL3
12	Solve using Master Theorem: $T(n) = 2T(n/2) + n^2$	KL3
13	Solve: $T(n) = T(n-2) + n$	KL3
14	Differentiate between Priori and Posteriori analysis	KL4
15	Explain the concepts of asymptotic notations and its properties.	KL2
	Unit-2	
	Short Question	
S.No	**Questions**	
1	Merge the following files optimally: (1, 3, 2, 5, 3, 4, 6, 8)	
2	Describe principle of optimality. With suitable example.	KL2
3	If an array is sorted in **decreasing order**, then which **sorting algorithm runs in minimal complexity**? Justify.	KL1
4	Describe what happens when the backtracking algorithm reaches a complete solution.	KL2
5	Describe the asymptotic worst-case running time for 0/1 Knapsack problem and Travelling Salesman Problem using Dynamic Programming?	KL2
6	Differentiate Greedy algorithm and Dynamic programming.	KL2
7	Draw state space tree of 4 queen problem	KL1
8	State Job Sequencing with Deadline Problem. Write down time and space complexity if problem solved by Greedy approach.	KL1
9	Why Greedy method is not suggested for 0/1 Knapsack problem?	KL1
10	Describe job scheduling with deadlines.	KL2
11	In which scenario a sorting technique is called stable? Also give example of **two stable sorting techniques**.	KL2
12	Merge the following files optimally: 5, 9, 4, 2, 1, 10.	KL2

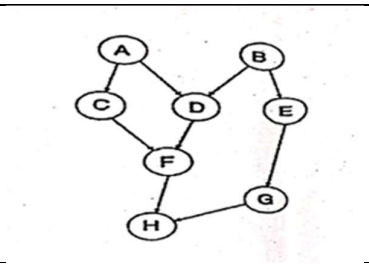
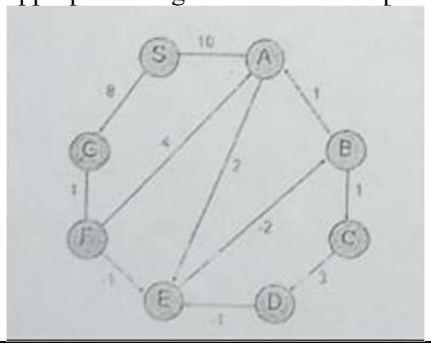
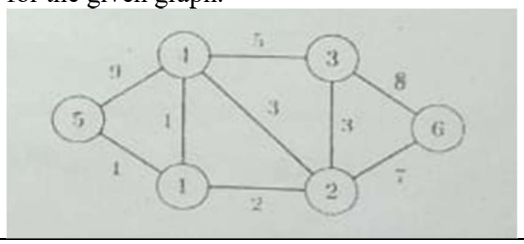
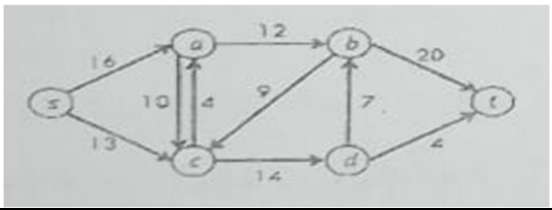
13	Differentiate **Divide and Conquer** and **Dynamic Programming** algorithm design technique. Justify with appropriate example.	KL2
14	Define principle of optimality. In which technique is it used?	KL1
15	State which of the following sorting algorithm are stable: insertion sort, merge sort or quick sort? And how?	KL1
16	Classify implicit and explicit constraints in backtracking methods.	KL2
17	Describe least cost search in branch and bound? Write the equation for the same.	KL2
18	Describe **Least Cost Search function** in brief.	KL2
19	Explain explicit and implicit constraints for **Hamiltonian Cycle**.	KL2
20	Explain Least Cost Search function for branch and bound algorithm design technique.	KL2
	Long Questions	
1	Define Merge–Purge rule with an appropriate example.	KL1
2	Solve the knapsack problem (0/1 and Fractional) using Greedy method: m = 30 $(w_1, w_2, w_3, w_4) = (10, 15, 6, 9)$ $(p_1, p_2, p_3, p_4) = (2, 5, 8, 1)$	KL3
3	Write the algorithm for Quick Sort and compute its time complexity. Also sort the following array using Quick Sort: A = (10, 25, 6, 34, 21, 76)	KL3
4	Write the backtracking algorithm for **N-Queen problem**. Find a solution to place **4 queens on a 4×4 chessboard**.	KL3
5	Solve the following **Travelling Salesman Problem using Branch and Bound**:  0 7 3 12 8 3 0 6 14 9 5 8 0 6 18 9 3 5 0 11 18 14 9 8 0	KL3
6	Explain Strassen's Matrix Multiplication in brief.	KL2
7	Write short note on Branch and Bound Technique	KL2
8	Write the algorithm for general iterative backtracking method and explain various factors that define the efficiency of backtracking.	KL2
9	Discuss greedy algorithm to generate single-source shortest path. Explain with example. Also analyze time complexity.	KL2
10	Describe the algorithm to find minimum-cost binary search tree. Show that the computing time of function OBST is $O(n^3)$.	KL2
11	Consider the following instance of knapsack problem:  Item X1 X2 X3 X4 X5 Profit 15 12 9 16 17 Weight 2 5 3 4 6 The maximum weight of 12 is allowed in the knapsack. Find the value of maximum profit with the optimal solution of the **fractional knapsack problem**.	KL3
12	Give a dynamic programming solution to the 0–1 knapsack problem that runs in O(nW) time, where n is the number of items and W is the maximum weight of items that can be put in the knapsack. Consider the weights and values of items listed below. The task is to pick a subset of these items such that their total weight is no more than 5 kg	KL3

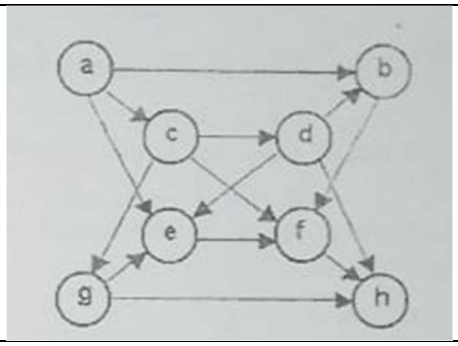
	and their total value is maximized. Item No. | Weight (Kg) | Values (Rs.) 1 | 2 | 3 2 | 3 | 4 3 | 4 | 5 4 | 5 | 6	
13	**(a)** The N-Queen problem is a classic problem in computer science, where the goal is to place **N queens on an N × N chessboard** so that no two queens attack each other. **(i)** Write a brute-force algorithm to solve the N-Queen problem. Analyze the time complexity of your algorithm, and explain why it is not efficient for large values of N. **(ii)** Write a backtracking algorithm to solve the N-Queen problem. Analyze the time complexity of your algorithm, and compare it with the brute-force algorithm.	KL3
14	Describe the Travelling Salesman Problem and explain why it is NP-complete.	KL2
15	Write the algorithm for **divide and conquer based sorting technique** which runs with the **same time complexity in all scenarios** and also compute the **time complexity of the algorithm**.	KL3
16	Compute the **shortest path from A to F using greedy method**. 	KL3
17	Write **Ford–Fulkerson Network Flow algorithm** and compute the **maximum unit flow from node 1 to node 8**. 	KL3
18	Solve the **knapsack problem using Branch and Bound technique**: $m = 30$ $(w_1, w_2, w_3, w_4) = (18,15,8,12)$ $(p_1, p_2, p_3, p_4) = (36,24,8,12)$	KL3
19	Differentiate **backtracking** and **dynamic programming** algorithm design techniques corresponding to their application domain. Also explain the **N-Queen problem in detail using backtracking**.	KL4
20	Explain job sequencing with deadlines. Generate a solution using the same for the following jobs. Job | A | B | C | D | E Profit | 100 | 19 | 27 | 25 | 15 Deadline | 2 | 1 | 2 | 1 | 3	KL2
21	Find the product of these two matrices using **Strassen's matrix multiplication method**. Derive the **time complexity of Strassen's matrix multiplication**.	KL3

	$A = \begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix} B = \begin{bmatrix} 8 & 4 \\ 6 & 2 \end{bmatrix}$	
22	Explain N-Queen problem using backtracking.	KL2
23	Define branch and bound strategy. Solve the following 0/1 Knapsack problem using branch and bound strategy with knapsack weight equal to 15.  	KL3
24	Briefly explain Multistage graphs with suitable examples.	KL2
25	Explain the general method of branch and bound.	KL2
	Unit-3	
	Short Question	
S.No	**Questions**	
1	Explain explicit and implicit constraints for Hamiltonian cycle.	KL2
2	Discuss why does Dijkstra's algorithm fail on negative weight?	KL2
3	Explain **Transitive Closure** in graph.	KL2
4	Differentiate **Binary Tree**, **Binary Search Tree**, and **Optimal Binary Search Tree**.	KL2
5	Differentiate Breadth First Search (BFS) and Depth First Search (DFS) graph traversal algorithms.	KL2
6	Describe Bi-connected components.	KL2
7	List some applications of Minimum Spanning Tree (MST).	KL1
8	Find out the topological orderings of the following directed acyclic graph (Assume starting vertex 1)  	KL1
9	Draw Binary Search Trees for the given set of keys and their corresponding frequencies and find the Optimal among them. keys[] = {10, 12, 20} freq[] = {34, 8, 50}	KL1
10	Bellman–Ford Algorithm always find negative weight cycle in the graph. Justify your answer.	KL1
	Long Questions	
1	Explain minimum spanning tree with its types.	KL2
2	Discuss Depth First Search (DFS) algorithm for graph traversal. Consider the following graph and find out the strongly connected components in it.  	KL2
3	The problem of determining whether there exists a cycle in an undirected graph is in P, NP, or both. Justify your answer.	KL2
4	Define spanning tree. Compute a minimum cost spanning tree for the given graph using **Prim's algorithm**. Also analyze its complexity.	KL3

		
5	Explain Bellman ford Algorithm.	KL2
6	Consider the given graph. In what order will the nodes be visited using a Breadth First Search and Depth First Search? (Assume starting vertex A.) 	KL2
7	A delivery truck travels between multiple destinations. The truck starts at city **A** and visits cities **B, C, D, and E** before returning to **A**. Design an algorithm to find the **shortest path for the truck to travel while minimizing cost**. Use the following distance matrix to solve the problem. 	KL3
8	Explain the general method of branch and bound.	KL2
9	Describe Topological Sorting. Explain with example.	KL2
10	In a given directed graph with source *s* and sink *t*, where each edge in the graph has a non-negative capacity, find the maximum flow that can be sent from *s* to *t*. 	KL3
11	If A to H are different steps of a task and their dependency is given in the graph below. Compute the **topological order of events** for successful completion of the task.	KL3

	A	B	C	D	E
A	0	10	20	15	30
B	10	0	25	30	20
C	20	25	0	35	15
D	15	30	35	0	10
E	30	20	15	10	0

		
12	Amit resides at city A and plans to visit various cities B, C and D. Compute the optimal route with minimum cost to return back to A and every city he wants to visit once. 	KL3
13	Explain Dijkstra algorithm. How is it different from Bellman–Ford algorithm? Apply the appropriate single source shortest path algorithm for the following graph. 	KL3
14	Explain Prim's algorithm. Apply Prim's algorithm to obtain a minimum cost spanning tree for the given graph. 	KL3
15	What are **residual networks, augmenting paths, and cuts** of a flow network? Describe the **Ford–Fulkerson algorithm** and solve the following graph for **maximizing the network flow**. 	KL3
16	**Find all topological sorting orderings of a given Directed Acyclic Graph (DAG) starting from ……**	KL3

		
17	Write short note on Hamiltonian Cycle.	KL2
18	Write short note on Floyd Warshalls Algorithm.	KL2
19	Write Network Flow algorithm and explain with appropriate example.	KL2
20	Define spanning tree. Write Kruskal's algorithm for finding minimum cost spanning tree. Describe how Kruskal's algorithm is different from Prim's algorithm.	KL2
	Unit-4	
	Short Question	
S.No	**Questions**	
1	Differentiate NP-Hard and NP-Complete problems.	KL2
2	Differentiate P and NP problems.	KL2
3	Differentiate **P, NP-Hard and NP-Complete problems**.	KL2
4	Differentiate between P, NP, NP-hard and NP-complete problems.	KL2
5	Classify deterministic and nondeterministic algorithms.	KL2
6	State and explain Cook's theorem	KL1
7	How are P and NP problems related.	KL2
8	Differentiate Time Efficiency and Space Efficiency.	KL2
9	Define the class NP. Give suitable examples.	KL1
10	Discuss the importance of NP-completeness in algorithm design.	KL2
	Long Questions	
1	Define **Cook's Theorem** and explain reduction of NP-Hard problems into **NP-Complete problem** with suitable example.	KL2
2	Write short note on NP-complete and NP-hard.	KL2
3	What is the relationship among P, NP, NP-Hard and NP-Complete problems? Show with the help of a diagram.	KL2
4	Explain the class of P and NP with example.	KL2
5	Explain the satisfiability problem and write the algorithm.	KL2
6	Explain halting problem explain with an example.	KL2
7	Compare NP-hard and NP-completeness.	KL4
8	Explain Vertex Cover problem and its NP-completeness.	KL2
9	What is polynomial time reduction? Explain with example.	KL2
10	Explain Clique problem and show that it is NP-complete.	KL2
	Unit-5	
	Short Question	
S.No	**Questions**	
1	Discuss why randomized quick sort is more preferable to normal quicksort?	KL2
2	Describe the importance of approximation algorithm	KL2
3	State the use of approximation algorithms.	KL1
4	Write is the role of randomness in algorithm design.	KL2
5	Define absolute and relative error in approximation algorithms.	KL2
6	Define an approximation algorithm. Why is it needed?	KL1
7	Discuss the role of randomness in algorithm design?	KL2
8	Define failure probability. How can it be reduced.	KL1

9	Define approximation ratio (performance guarantee).	KL1
10	Describe the concept of inapproximability.	KL2
	Long Questions	
1	Explain Approximation Algorithms in detail.	KL2
2	Explain randomized algorithms in detail.	KL2
3	Explain **any two types of Randomized algorithm techniques** with corresponding applications and scope.	KL2
4	Explain the **utility of Approximation algorithms** with appropriate example.	KL2
5	Write an algorithm for randomizes quick sort. Find its worst-case time complexity.	KL2
6	Write short note on Las Vegas and Monte Carlo Algorithm	KL2
7	Write the advantage of randomized algorithms over deterministic ones.	KL2
8	Explain the approximation algorithm for Traveling Salesman Problem (TSP).	KL2
9	Difference between exact algorithm and approximation algorithm	KL4
10	Compare randomized algorithms and approximation algorithms.	KL4