Roll Number:_____

Thapar Institute of Engineering & Technology

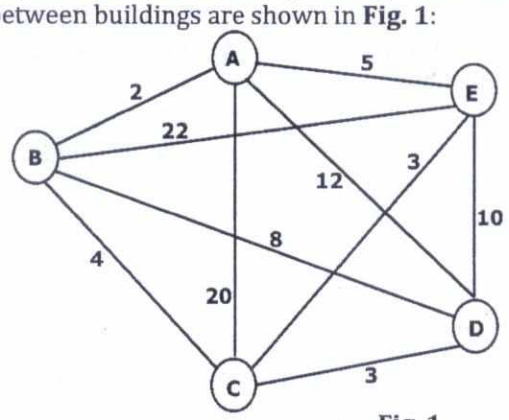
Department of Computer Science and Engineering

END SEMESTER EXAMINATION

B. E. (2nd Yr. COE/CSE)
30th May, 2025
Friday, Time– 2:00 PM To 5:00 PM
Time: 3 Hours, **Max Marks: 75+5*** (Weightage: 40)
***5 marks for adhering to the instructions**

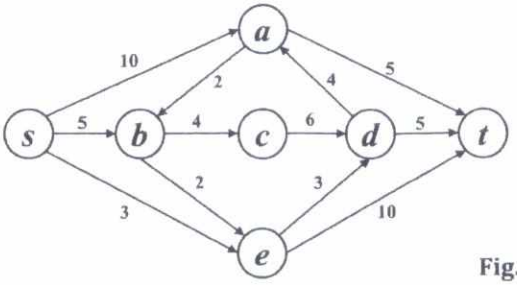
Course Code: UCS415
Course Name: Design and Analysis of Algorithms
Name of Faculty: Tarunpreet Bhatia, Sumit Sharma, Shubhra Dwivedi, Nisha Chauhan, Ritesh Sharma, Kapil Tomar, Neeraj Sharma, Shruti Aggarwal

Instructions: Attempt any 3 Questions. Answer all sub-parts of each question at one place. Do mention Page No. of your attempt at front page of your answer sheet. Assume missing data (if any).

Q. No	Questions	Marks	CO	BL
Q.1	(a) A distributed database system optimizes complex queries by dividing them into multiple equal-sized subqueries, sending each to different nodes. Each subquery reduces the problem size by one third, and all are processed recursively. After collecting results, a merge and aggregation step is performed, taking time proportional to the input size. **Write the recurrence relation** and **analyze** the time complexity of this approach using the **Master Method.**	(5)	CO1	L4
	(b) University campus has 5 buildings numbered A through E. Shuttle service needs to create an efficient route that begins at building A, visits each of the other four buildings exactly once, and returns to building A. The travel times (in minutes) between buildings are shown in **Fig. 1**:		CO4	L3
	(i) Draw the **state space tree** for the Traveling Salesman Problem (TSP) and compute **optimal time** and shuttle tour using **backtracking.**	(5)		
	(ii) Determine the **optimal time** required for the shuttle tour using **branch and bound**. Show the intermediate steps and draw the **state space tree**. **Note**: At any level of state space tree, if cost of 2 nodes is same then choose the one which is alphabetically (lexicographically) smaller.	(10)		
	(iii) Calculate the **approximate time** needed for the shuttle tour using **2-approximation algorithm** and determine the **approximation ratio.**	(5)		
	 **Fig. 1**			
Q.	(a) Given a text T[1. . .n] and a pattern P[1. . .m], **write an algorithm** that prints all occurrences of P[] present in T[] using **Rabin Karp** algorithm. You may assume that n > m. **Apply** your algorithm to text **T = "DAACABCDBA"** and a pattern **P = "DBA"**. Consider input alphabet size as 26 and assign numerical values to characters in input set {A,B,C,D} as {1,2,3,4}. Use modulus q as 11.	(3+7)	CO3	L2, L3
	(b) The City Food Bank is organizing a relief distribution drive for a flood-affected community. They have two delivery trucks, each with a maximum carrying capacity of 7 tons. The food bank has collected five types of essential food items, each differing in weight and nutritional value (in nutrition units) as shown the below Table:		CO2	L3, L4

Item	1	2	3	4	5
Description	Rice Bags	Flour	Cooking Oil	Canned Vegetables	Protein Bars
Weight (ton)	2	3	1	4	2
Nutritional Value(units)	1700	2100	900	2300	1000

(i) **Truck A** uses standard packaging and can carry food items in their complete original packages due to packaging and logistics constraints. Each item can either be fully loaded or not loaded at all. **Determine the optimal combination** of items to load into Truck A to maximize total nutritional value without exceeding 7 tons using dynamic programming (DP). **Write** the **recursive equation** and **fill the DP table values** only.	(8)		
(ii) **Truck B** uses flexible packaging that allows it to carry partial quantities of the food items (i.e. Items can be divided). **Determine the optimal allocation of items (including partial items)** for Truck B to maximize total nutritional value within the 7 tons limit.	(4)		
(iii) **Compare** the **time complexities** of both packaging strategies and **calculate** the difference in total nutritional value between the two strategies.	(3)		

Q.3	(a) Explain the significance of the residual network in the **Ford-Fulkerson algorithm**. Consider a Flow Network shown in **Fig. 2.** You need to **compute a maximum flow** in this network using the Ford-Fulkerson algorithm from source s to destination t. For each iteration, draw residual network, identify the augmenting path, its bottleneck/residual capacity, and the value of the flow at the end of the iteration. Also, **find a cut** in the network whose capacity is equal to the value of the flow as computed by you. Is the cut you found the minimum cut? If not, identify the actual minimum cut of the given flow network.  **Fig. 2**	(2+10 +3)	CO3	L2, L3
	(b) A major e-commerce company aims to optimize the transmission of product descriptions across its network by compressing the text data. To achieve this, the company has collected frequency statistics for the characters commonly used in these descriptions. The objective is to reduce the total number of bits needed for transmission. The character frequencies are as follows: 'A' appears 50 times, 'B' 30 times, 'C' 20 times, 'D' 10 times, 'E' 40 times, and 'F' 15 times. (i) Construct the **Huffman tree** and **determine** the binary codes for each character. (ii) Huffman coding always results in a prefix code. Why is the **prefix property** important in data compression and decoding? **Illustrate** with the help of an **example**. (iii) Calculate the **percentage saving** in average code length achieved by Huffman encoding compared to fixed-length encoding (3 bit each character code).	(6) (2) (2)	CO2	L3 L4
Q.4	(a) You are given N identical eggs and a K floor building (floors are numbered from 1 to K). There exists a critical floor F $(0 \leq F \leq K)$ such that: • Any egg dropped from above F will break. If the egg breaks on a certain floor, it will break on any floor above. • Any egg dropped from or below F will not break. If the egg doesn't break at a certain floor, it will not break at any floor below. • An egg that survives a fall can be used again and a broken egg must be discarded. The minimum number of trials needed in the worst case for this critical floor F can be determined using the following dynamic programming recurrence equation: $$F[N,K] = \begin{cases} K, & if\ N == 1\ or\ K == 1\ or\ K == 0 \\ 1 + min_{x=1}^{K}\left\{ \max\begin{pmatrix} F[N-1,x-1], \\ F[N,K-x] \end{pmatrix} \right\}, & otherwise \end{cases}$$ You are just required to **fill the DP table values** for $N = 4$ eggs and $K = 7$ floors using this recurrence equation and **find the minimum number of trials** needed for this critical floor F.	(10)	CO1	L3
	(b) A software company is managing a complex application consisting of several interdependent modules (labelled A to J). The development team notices that the build process sometimes fails due to circular dependencies among certain modules, while others compile successfully in a clear order. To investigate the issue, the team visualizes the module dependencies as a directed graph. In this graph, each module is a node, and an arrow from Module A to Module B means B depends on A. The dependency relationships are as follows: A→B, A→H, B→B, B→C, C→B, C→F, D→C, D→E, E→J, F→G, F→J, G→C, H→A, H→G, H→I, I→G, I→J, J→E (i) The team needs to identify groups of modules that are tightly coupled and form a mutual dependency and cannot be compiled independently. **Design an algorithm and apply your algorithm** to identify such groups of modules for given module dependencies. Show intermediate steps and draw the resultant condensed graph. (ii) Based on your analysis and condensed graph, **reorganize the system** so that the resultant modules can be compiled in a sequence that respects all dependencies. Show the steps to organize the build sequence.	(3+7) (5)	CO4	L6