Program	Bachelor of Technology (B.Tech.) CSE	**Semester-4**
Type of Course	Professional Core	
Prerequisite	Data Structure	
Course Objective	This course introduces various methods to design and analyze algorithms. Students will learn different algorithms for given computational tasks and evaluate their relative merits based on the performance measures.	

Teaching Scheme				Examination Scheme				
				Theory Marks		Practical Marks		Total Marks
Lecture	Tutorial	Lab	Credit	SEE	CIA	SEE	CIA	
3	0	2	4	70	30	25	25	150

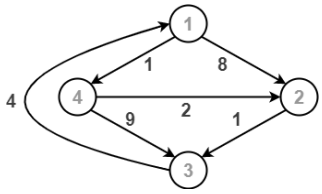
List of Practical

1.	A	**Write following programs and measure the time of execution: (Basics of Algorithms and Mathematics)** 1. Calculate the Factorial of N number. 2. Calculate the sum of 1 to N number. 3. Print first 50 natural numbers 4. Calculate x^y without using power function. 5. Implement the programs 1, 2, 3 and 4 using Recursive method.
2.	A	**Write following programs and measure the time of execution: (Basics of Algorithms and Mathematics)** 1. Generate Fibonacci series up to n terms using iterative method. 2. Print the array elements (Read the elements from file). 3. Print the sum of the digits of a given number. 4. Implement the programs 1, 2 and 3 using Recursive method.
3.	A	**Write following programs: (Basics of Data Structure)** 1. Implement stack operations (PUSH, POP, PEEP, CHANGE & DISPLAY) 2. Implement queue operations (INSERT, DELETE, DISPLAY) 3. Implement singly linked list operations (INSERT, DELETE, DISPLAY)
4.	A	**Analysis of Algorithm** 1. Write an algorithm of Bubble sort and sort the following data using Bubble sort method • 65, 77, 5, 25, 32, 45, 99, 83, 69, 81 • U, N, I, V, E, R, S, E 2. Write a program to sort a given set of elements using Bubble sort. 3. Write an algorithm of Insertion sort and sort the following data using Insertion sort method • 12, 10, 45, 8, 19, 25, 27, 20, 44 • E, D, U, C, A, T, I , O, N 4. Write a program to sort a given set of elements using Insertion sort.
	B	1. Determine the time required to sort the element using Bubble sort. (input data from text file)
5.	A	**Analysis of Algorithm** 1. Write an Algorithm of Selection sort and Sort the following data using Selection sort method • 20, 50, 30, 75, 90, 60, 25, 10, 40 • 12, 10, 45, 8, 19, 25, 27, 20, 44 2. Write an Algorithm of Heap sort and Sort the following data using Heap sort method. • 65, 77, 5, 25, 32, 45, 99, 83, 69, 81 3. Write a program to sort a given set of elements using Selection sort. 4. Sort the given numbers using Counting algorithm. • 1, 4, 1, 2, 7, 5, 2
	B	1. Write a program to sort a given set of elements using Heap sort.

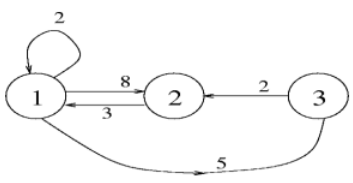
		2. Write a program to sort a given set of elements using Counting sort.
6.	A	**Divide and Conquer Algorithm** 1. Write a program to implement Linear Search algorithm. 2. Write a program to implement Binary Search algorithm. 3. Using divide and conquer multiply the following numbers. • u = 2345 and v = 5678 • u = 1026 and v = 7329 • u = 1026732 and v = 732912
	B	1. Write a program to perform large integer multiplication using divide and conquer.
7	A	**Divide and Conquer Algorithm** 1. Write an algorithm of Quick Sort Method and sort given numbers using Quick Sort algorithm. • 3, 1, 4, 5, 9, 2, 6, 5 • 6, 5, 3, 11, 10, 4, 7, 9 2. Write an algorithm of Merge Sort Method and sort given data using Merge Sort algorithm. • 12, 31, 25, 8, 32, 17, 40, 42 • G, U, J, A, R, A, T 3. Write a program to sort a given set of elements using Quick sort.
	B	1. Write a program to sort a given set of elements using Merge sort.
8	A	**Dynamic Programming** 1. Solve making change problem using Dynamic programming.

Sr.	Denominations	Making change of Rs.
1	d1= 1, d2=3, d3=5, d4=6	8
2	d1= 1, d2=5, d3=6, d4=8	11
3	d1= 1, d2=4, d3=10	13
4	d1= 1, d2=2, d3=3, d4=5	9

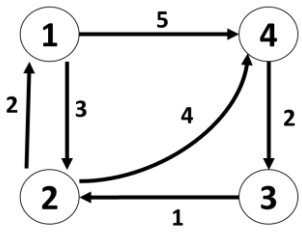
		2. Derive an equation for solving the 0/1 Knapsack problem using dynamic programming method. Solve the following instance of knapsack problem using Backtracking technique. • W = 8 and w = (2,3,4,5) and value v = (3,5,6,10) • W = 9 and w = (3,4,5,7) and value v = (12,40,25,42) • W = 10 and w = (4,2,3,1,6,4) and value v = (6,4,5,3,9,7)
	B	1. Write a program to Implement 0/1 knapsack problem using dynamic method
9	A	**Dynamic Programming** 1. Solve all pair shortest path problem for the following given graph using Floyd's algorithm.



(I)

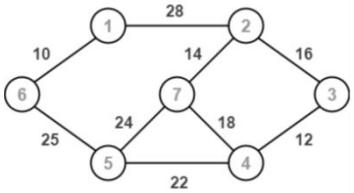
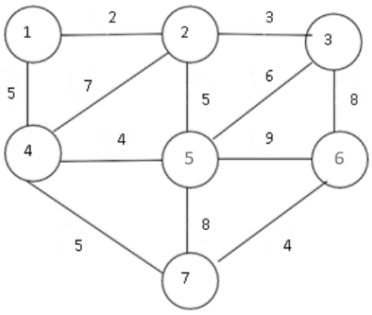
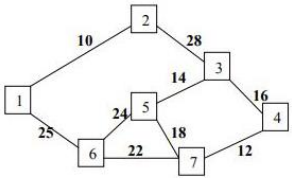
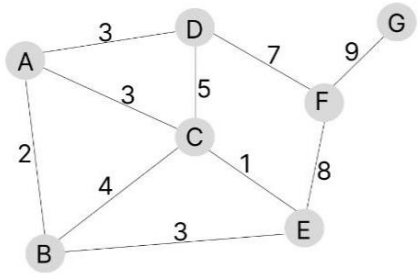
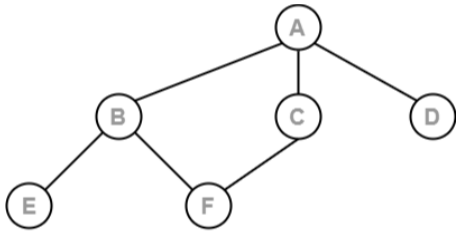


(II)



(III)

		2. Using dynamic programming find out the optimal sequence for the matrix chain multiplication • A [5x4], B [4x6], C [6x2] and D [2x7] matrices • A [4x10], B [10x3], C [3x12], D [12x20] and E [20x7] matrices • A1[10x100], A2[100x5], A3[5x50] and A4[50 x1] matrices 3. Given two sequence of characters, Obtain the longest common subsequence.

		• X={G,U,J,A,R,A,T} and Y = {J,R,A,T} • A={K,A,N,D,L,A,P} and B = {A,N,D,L} • X=abbacdcba and Y=bcdbbcaac
	B	1. Write a program to implement making a change problem using dynamic programming.
10	A	**Greedy Algorithm** 1. Using Activity selection problem, Solve the following • I1 (1-3), I2 (0-2), I3 (3-6), I4 (2-5), I5 (5-8), I6 (3-10), I7(7-9) • I1 (1-3), I2 (0-4), I3 (1-2), I4 (4-6), I5 (2-9), I6 (5-8), I7(3-5), I8(4-5) 2. Find Minimum Spanning Tree for the given graph using Prim's Algorithm.  (I)  (II)  (III) 3. Find Minimum Spanning Tree for the above graphs using Kruskal's algorithm
	B	1. Write a program to study and implement minimum spanning tree using Kruskal's algorithm. 2. Write a program to study and implement minimum spanning tree using Prim's algorithm.
11	A	**Greedy Algorithm** 1. Solve the following Knapsack Problem using greedy method • W = 20 and w = (18,15,25) and value v = (25,24,15) • W = 15 and w = (2,3,5,7,1,4,1) and value v = (10,5,15,7,6,18,3) • W = 40 and w = (20,25,10) and value v = (30,40,35) 2. Using greedy algorithm find an optimal schedule for following jobs • n=6. Profits: (20,15,10,7,5,3) Deadline: (3,1,1,3,1,3) • n=7. Profits: (35,30,25,20,15,12,5) Deadline: (3,4,4,2,3,1,2) • n=5. Profits: (20,15,10,5,1) Deadline: (2,2,1,3,3)
	B	1. Write a program to implement job sequencing algorithm using greedy approach.
12	A	**Exploring Graphs** 1. Traverse the following graph using BFS  (I)  (II) 2. Traverse the above graphs using DFS.

	B	1. Write a C program to implement BFS algorithm.
13	A	**String Matching** 1. Show the comparisons the naive string matcher makes for the pattern P=0001 in the text T=000010001010001. 2. Solve the given example using Rabin Karp matcher. • Working modulo q=11, text T= 3141592653589793, pattern P = 26 • Working modulo q=13, text T = 2359023141526739921, pattern P = 31415
	B	1. Write a C program to implement Rabin-Karp method for pattern searching