Program	BACHELOR OF TECHNOLOGY (B.Tech)	**Semester - 4**
Type of Course	Professional Core Course	
Prerequisite		
Rationale	-	
Effective From A.Y.	2025-26	

Teaching Scheme (Contact Hours)				Examination Scheme				
Lecture	Tutorial	Lab	Credit	Theory Marks		Practical Marks		Total Marks
				SEE TH	IAT	SEE P	CCE	
3	0	2	4	70	-	50	-	200

***SEE** - Semester End Examination, **IAT** - Internal Assessment Test, **CCE** - Continues & Comprehensive Evaluation*

Course Content

T - Teaching Hours | **W** - Weightage

Sr.	Topics	T	W
1	**Analysis of Algorithms:** The efficient algorithm, Average, Best and Worst case analysis, Amortized analysis, Asymptotic Notations, Analyzing control statement, Solving recurrences, Sorting Algorithms and analysis: Selection Sort, Insertion Sort, Shell Sort, Heap sort, Sorting in linear time: Bucket sort, Radix sort and Counting sort.	9	20
2	**Divide and Conquer Strategy:** Introduction, Multiplying large integers problem, Problem Solving using divide and conquer algorithm - Binary Search, Sorting (Merge sort, Quick sort), Strassen's matrix multiplication, Exponential.	4	10
3	**Greedy Algorithms** General Characteristics of greedy algorithms, Elements of greedy strategy, Making change problem, Fractional-Knapsack problem, Job Scheduling Problem, Huffman code, Activity selection problem, Tape storage problem.	9	9
4	**Dynamic Programming** Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming – Calculating the Binomial Coefficient, Memorization, Making Change Problem, Assembly Line-Scheduling, Knapsack problem, All Points Shortest path, Matrix chain multiplication, Longest Common Subsequence.	9	20
5	**Exploring Graph and Backtracking:** Articulation points, Topological sort, Strongly connected components, Maximum bipartite cover problem, Network flows: Ford Fulkerson algorithm, Max-flow Min-cut theorem, Eight queen problem, Travelling salesman problem, Branch and bound - Assignment problem, Knapsack problem.	9	20
6	**String Matching and NP-Completeness:** String matching applications, Naive string-matching algorithm, Rabin-Karp algorithm, String matching with finite automata, The Knuth-Morris-Pratt algorithm, The class P and NP, Polynomial reduction, NP- Completeness Problem, NP-Hard Problems. Travelling Salesman problem, Approximation algorithms, Randomized algorithms.	5	10
	Total	**45**	**89**

Suggested Distribution Of Theory Marks Using Bloom's Taxonomy

Level	Remembrance	Understanding	Application	Analyze	Evaluate	Create
Weightage	20	30	40	10	0	0

NOTE : This specification table shall be treated as a general guideline for the students and the teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcomes

At the end of this course, students will be able to:	
CO1	Analyze the efficiency of algorithms by evaluating and linking various comparison-based linear-time sorting algorithms.
CO2	Discuss various searching and sorting problems using divide and conquer techniques.
CO3	Understand and analyze the performance of algorithms for different problems using greedy approach.
CO4	Explain the dynamic-programming paradigm, identify situations where it is the appropriate design approach, and describe and analyze algorithms that utilize this paradigm.
CO5	Implement major graph algorithms and their analysis, use graphs to model engineering problems, and develop new graph-based solutions while also demonstrating algorithms using branch and bound and backtracking techniques.
CO6	Comprehend various fundamental algorithms for quickly performing string matching and understand the concepts of intractable problems.

CO PO Mapping

CO	CO - 1	CO - 2	CO - 3	CO - 4	CO - 5	CO - 6
PO - 1						
PO - 2						
PO - 3						
PO - 4						
PO - 5						
PO - 6						
PO - 7						
PO - 8						
PO - 9						
PO - 10						
PO - 11						

Reference Books	
1.	Introduction to Algorithms", Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein- 4th Edition, MIT, 2022 (TextBook)
2.	Design and Analysis of Algorithms", A. V. Aho, J. D. Ullman – 2nd Edition, Pearson LPE, 2017
3.	Fundaments of Computer Algorithms", Horowitz Sahni, Rajasekaran – 3rd Edition, Universities Press, 2021
4.	Fundamental of Algorithms", Gills Brassard, Paul Bratley-2nd Edition, PHI, 2015. (TextBook)
5.	Design and Analysis of Algorithms", Parag H. Dave and Himanshu B. Dave –2nd Edition,2013. Pearson. (TextBook)

List of Practical	
1.	Implement following algorithm in iterative and recursive manner: A. a. GCD algorithm B. b. Factorial algorithm c. Fibonacci algorithm d. Tower of Hanoi
2.	Implement insertion sort algorithm and perform its best case, average case and worst-case analysis.
3.	Implement max-heap sorting algorithm and perform its best case, average case and worst-case analysis.
4.	Implement Quick sort algorithm using divide and conquer approach.
5.	Implement Merge sort algorithm using divide and conquer approach.
6.	Implement fractional knapsack problem using greedy approach.
7.	Implement activity selection problem using greedy approach.
8.	Implement making change problem using dynamic programming.
9.	Implement longest common subsequence problem using dynamic programming.
10.	Implement topological sorting algorithm.
11.	Implement travelling salesman problem using backtracking.
12.	Implement string matching problem using naïve string-matching algorithm.
13.	Implement string matching problem using Rabin- Karp algorithm.