Course: BTech **Semester:** 5

Prerequisite: Data structures, Fundamental of programming

Rationale: Analyze the asymptotic performance of algorithms. Write rigorous correctness proofs for algorithms. Demonstrate a familiarity with major algorithms and data structures. Apply important algorithmic design paradigms and methods of analysis. Synthesize efficient algorithms in common engineering design situations.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction and Analysis of Algorithms:** Algorithm: Definition, Properties, Types of Algorithms, Writing an AlgoritAlgorithm Analysis: Parameters, Design Techniques of Algorithms Asymptotic Analysis: Big Oh, Big Omega & Big Theta Notations, Lower Bound, Upper Bound and Tight Bound, Best Case, Worst Case, Average Case Analyzing control statement, Loop invariant and the correctness of the algorithm, Recurrences- substitution method, recursion tree method, master method. Sorting Techniques with analysis: Bubble Sort, Selection Sort, Insertion sort.	20	10
2	**Divide & Conquer Algorithms:** Structure of divide-and-conquer algorithms, examples: Binary search, quick sort, Merge sort**,** Strassen Multiplication; Max-Min problem	20	6
3	**Greedy Algorithms:** Introduction, Elements of Greedy Strategy - Minimum Spanning Tree: Kruskal's & Prim's Algorithm, Dijkstra’s Algorithm, Knapsack Problem, Activity Selection Problem, Huffman Codes	20	8
4	**Dynamic Programming:** Principal of Optimality, 0/1 Knapsack Problem, Making Change problem, Chain matrix multiplication, Longest Common Subsequence, All pair shortest paths: Warshall’s and Floyd’s algorithms	20	8
5	**Exploring Graphs:** An introduction using graphs and games, Undirected Graph, Directed Graph, Traversing Graphs, Depth First Search, Breath First Search, Topological sort	5	3
6	**Backtracking and Branch & Bound:** Introduction to Backtracking, Introduction to Branch & Bound, 0/1 Knapsack Problem, N-Queens Problem, Travelling Salesman Problem	5	4
7	**String Matching & NP Completeness:** **String Matching:** - Introduction to String Matching, Naive String Matching, Rabin-Karp Algorithm, Kruth-Morris-Pratt Algorithm, String Matching using Finite Automata **NP Completeness:** - Introduction to NP Completeness, P class Problems, NP Class Problems, Hamiltonian Cycle	10	6

Reference Books

1.	**Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill. (TextBook)**
2.	**Fundamentals of Algorithms – E. Horowitz et al. (TextBook)**
3.	**Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson**
4.	**Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.**
5.	**Algorithms—A Creative Approach,3RD Edition, UdiManber, Addison-Wesley, Reading, MA**

Course Outcome

After Learning the Course the students shall be able to:

Course Outcome: After learning the course the students will be able to:
1. Develop the ability to analyze the running time of any given algorithm using asymptotic analysis and prove the correctness of basic algorithms.
2. Design efficient algorithms for computational problems, using various algorithm design techniques taught in the course.
3. Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.
4. Analyze String matching algorithms.
5. Explain the complexity classes P, NP, and NP-Complete, and demonstrate the NP-Completeness of a specific problems.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc