

Maulana Abul Kalam Azad University of Technology, West Bengal*(Formerly West Bengal University of Technology)***Syllabus for B. Tech in Computer Science & Engineering**

(Applicable from the academic session 2018-2019)

Advanced Algorithms**Code: PEC-IT601 A****Contact: 3L**

Name of the Course:	Advanced Algorithms	
Course Code: PEC-IT601A	Semester: VI	
Duration:6 months	Maximum Marks:100	
Teaching Scheme	Examination Scheme	
Theory:3 hrs./week	Mid Semester exam: 15	
Tutorial: NIL	Assignment and Quiz: 10 marks	
	Attendance: 5 marks	
Practical: NIL	End Semester Exam:70 Marks	
Credit Points:	3	
Objective:		
1	Introduce students to the advanced methods of designing and analyzing algorithms.	
2	The student should be able to choose appropriate algorithms and use it for a specific problem.	
3	To familiarize students with basic paradigms and data structures used to solve advanced algorithmic problems.	
4	Students should be able to understand different classes of problems concerning their computation difficulties.	
5	To introduce the students to recent developments in the area of algorithmic design.	
Pre-Requisite:		
1	Algorithm Design and Analysis	

Unit	Content	Hrs/Unit	Marks/Unit
1	Sorting: Review of various sorting algorithms, topological sorting Graph: Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkstras), depth-first search and computation of strongly connected components, emphasis on correctness proof of the algorithm and time/space analysis, example of amortized analysis.	6	
2	Matroids: Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set. Application to	8	

	MST. Graph Matching: Algorithm to compute maximum matching. Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path.		
	Flow-Networks: Maxflow-mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm. Matrix Computations: Strassen's algorithm and introduction to divide and conquer paradigm, inverse of a triangular matrix, relation between the time complexities of basic matrix operations, LUP-decomposition.	9	
3	Shortest Path in Graphs: Floyd-Warshall algorithm and introduction to dynamic programming paradigm. More examples of dynamic programming. Modulo Representation of integers/polynomials: Chinese Remainder Theorem, Conversion between base-representation and modulo-representation. Extension to polynomials. Application: Interpolation problem. Discrete Fourier Transform (DFT): In complex field, DFT in modulo ring. Fast Fourier Transform algorithm. Schonhage-Strassen Integer Multiplication algorithm	10	
4.	Linear Programming: Geometry of the feasibility region and Simplex algorithm NP-completeness: Examples, proof of NP-hardness and NP-completeness. One or more of the following topics based on time and interest Approximation algorithms, Randomized Algorithms, Interior Point Method, Advanced Number Theoretic Algorithm	10	
5	Recent Trends in problem solving paradigms using recent searching and sorting techniques by applying recently proposed data structures.	5	

Text book and Reference books:

1. "Introduction to Algorithms" by Cormen, Leiserson, Rivest, Stein.
2. "The Design and Analysis of Computer Algorithms" by Aho, Hopcroft, Ullman.
3. "Algorithm Design" by Kleinberg and Tardos.
4. Design & Analysis of Algorithms, Gajendra Sharma, Khanna Publishing House, New Delhi

Course Outcomes:

On completion of the course students will be able to

1. Analyze the complexity/performance of different algorithms.
2. Determine the appropriate data structure for solving a particular set of problems.
3. Categorize the different problems in various classes according to their complexity.
4. Students should have an insight of recent activities in the field of the advanced data structure.