

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)

Design and Analysis of Algorithms

Code: PCC-CS404

Contacts: 3L

Name of the Course:	Design and Analysis of Algorithms		
Course Code: PCC-CS404	Semester: IV		
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: hrs./week		End Semester Exam: 70 Marks	
Credit Points:	3		
Objective:			
1	The aim of this module is to learn how to develop efficient algorithms for simple computational tasks and reasoning about the correctness of them		
2	Through the complexity measures, different range of behaviors of algorithms and the notion of tractable and intractable problems will be understood.		
Pre-Requisite:			
1	To know data-structure and basic programming ability		

Unit	Content	Hrs/Unit	Marks/Unit
1	Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem	8	
2	Fundamental Algorithmic Strategies: Brute-Force, Greedy, Dynamic Programming, Branch and-Bound and Backtracking methodologies for the design of algorithms; Illustrations of these techniques for Problem-Solving, Bin Packing, Knap Sack TSP. Heuristics –characteristics and their application domains.	8	
3	Graph and Tree Algorithms: Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm.	6	
	Tractable and Intractable Problems: Computability		

4.	of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Cook's theorem, Standard NP-complete problems and Reduction techniques.	10	
5	Advanced Topics: Approximation algorithms, Randomized algorithms, Class of problems beyond NP – P SPACE	4	

Text books/ reference books:

1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2. Fundamentals of Algorithms – E. Horowitz et al.
4. Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson.
5. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
6. Algorithms -- A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA
7. Design & Analysis of Algorithms, Gajendra Sharma, Khanna Publishing House (AICTE Recommended Textbook – 2018)
8. Algorithms Design and Analysis, Udit Agarwal, Dhanpat Rai

Course Outcomes

On completion of the course students will be able to

PCC-CS404.1 For a given algorithms analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms.

PCC-CS404.2 Describe the greedy paradigm and explain when an algorithmic design situation calls for it. For a given problem develop the greedy algorithms.

PCC-CS404.3 Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Synthesize divide-and-conquer algorithms. Derive and solve recurrence relation.

PCC-CS404.4 Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. For a given problems of dynamic-programming and

PCC-CS404.5 develop the dynamic programming algorithms, and analyze it to determine its computational complexity.

PCC-CS404.6 For a given model engineering problem model it using graph and write the corresponding algorithm to solve the problems.

PCC-CS404.7 Explain the ways to analyze randomized algorithms (expected running time, probability of error).

PCC-CS404.8 Explain what an approximation algorithm is. Compute the approximation factor of an approximation algorithm (PTAS and FPTAS).