

Subject Code: 1ET2030101	Subject Title: DESIGN AND ANALYSIS OF COMPUTER ALGORITHMS
Pre-requisite	Fundamentals of Data Structures and Algorithms

Course Objective:

The main objective of this course is to make the students aware regarding problem formulation and transforming human readable steps into suitable programming languages. The course also intends to prepare the students to design efficient algorithms, which is measured in two directions; viz. Time & Space. Study and comparison of various computing algorithm will assist the students to understand, analyze and design better algorithms.

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
Lecture	Tutorial	Practical	Credit	Theory		Practical		Total
				University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
2	-	2	4	60	40	30	20	150

Subject Contents				
Sr. No	Topic	Total Hours	Weight (%)	
01	Introduction: Problem solving approach, Basics of set theory, Functions and relations, Designing algorithms, Recursive mathematical definitions	2	8	
02	Analysis of Algorithms Asymptotic analysis: Analyzing algorithms, Asymptotic Notations, Standard complexity classes, Empirical measurements of performance, Time and space tradeoffs in algorithms.	3	13	
03	Divide and Conquer: Analyzing recursive algorithms using recurrence relations (The substitution method for solving recurrences, Master method), Strassen's algorithm for matrix multiplication	4	17	
04	Dynamic programming: Making change problem, The principle of optimality, The knapsack problem, Floyd's algorithm for shortest paths	3	13	
05	Greedy Algorithms: Making change problem, Knapsack problem, Shortest Path – Dijkstra's algorithm, Huffman codes	3	13	
06	Amortized analysis: Aggregate analysis, Accounting method, Potential method, Single source shortest paths, Bellman Ford.	3	12	
07	String matching: The naïve string matching, Introduction to NP completeness, The class P and NP, Polynomial reductions, NP complete problems	3	12	
08	Heuristic algorithm: The travelling salesperson, Approximate algorithms-knapsack problem	3	12	

Review Presentation: The student is expected to spend minimum 2 hours per week (as mentioned in the teaching and evaluation scheme) to refer at least two peer reviewed journal papers related to this domain/subject. The student is expected to identify issues/challenges and emerging trends in the domain/subject. Student is supposed to explore various video lectures (E.g. NPTEL) available in the domain/subject. Student is required to make a review-presentation on the work carried out for the same.

Recommended sites for journal papers are (1) dl.acm.org (2) springer.com (3) sciencedirect.com (4) elsevier.com (5) ieeexplore.ieee.org (6) scholar.google.co.in (7) scopus.com or others of similar repute.

Course Outcome:

Upon completion of this course, students will be able to:

- Analyze the asymptotic performance of 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.

List of References:

1. Introduction to Algorithms. Thomas Cormen, Charles Leiserson, Ronald Rivest. PHI publication
2. Fundamentals of Algorithms. Gilles Brassard, Paul Bratley. PHI publication
3. Introduction to Design and Analysis of Algorithms, Anany Levitin, Pearson.
4. Foundations of Algorithms, Shailesh R Sathe, Penram
5. Design and Analysis of Algorithms, Dave and Dave, Pearson.

E-Resources / Web Links

1. <http://www.nptel.ac.in/courses/106101059/>

List of Experiments:

Note: The experiment list provided beneath is for reference only. The course teacher may change/formulate it as per his/her methodology and requirement.

- 01 Implement max-heap sort algorithm.
- 02 Implement knapsack problem using dynamic programming.
- 03 Implement chain matrix multiplication using dynamic programming.
- 04 Implement making a change problem using dynamic programming.
- 05 Implement a knapsack problem using greedy algorithm.
- 06 Implement Graph and Searching (DFS and BFS).
- 07 Implement prim's algorithm.
- 08 Implement kruskal's algorithm.
- 09 Implement LCS problem.
- 10 Find longest common sequence for $X = (A, B, C, B, D, A, B)$ and $Y = (B, D, C, A, B, A)$ using dynamic programming.
- 11 Given integer $a_1, a_2, \dots$ find contiguous subsequence whose sum is maximum using any programming language. e.g. If input is $\{8, -4, 13, -2, 5\}$ then output is 20. If input is $\{5, -3, 4, -2, -1, 6\}$ then output is 9.
- 12 Given a set of n integers does it contain a pair of elements a, b such that $a+b=0$?
- 13 Given a set of n integers does it contain a pair of elements a, b, c such that $a+b=c$?