

<b>Course Title</b>	<b>Design &amp; Analysis of Algorithms</b>
<b>Course Code</b>	CE610
<b>Course Credit</b>	Theory : 03
	Practical : 01
	Tutorial : 00
	Credits : 04

**Course Learning Outcomes**

On the completion of the course, students will be able to:

- **Demonstrate** a familiarity with major algorithms and data structures.
- **Apply** important algorithmic design paradigms and methods of analysis.
- **Analyze** the asymptotic performance of algorithms.
- **Create** Write rigorous correctness proofs for algorithms.
- **Understand** Basic knowledge of computational complexity, approximation and randomized algorithms.
- **Understand and design** algorithms using greedy strategy, divide and conquer approach, dynamic programming, and max flow - min cut theory.
- **Synthesize** efficient algorithms in common engineering design situations
- **Understand** basic knowledge of graph and matching algorithms.
- **Analyze** lower-bounds and the theory of NP-completeness

Sr. No.	Name of chapter & details	Hours Allotted
<b>Section – I</b>		
<b>1</b>	<b>Introduction:</b> Introduction: What is an Algorithm?, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Fundamental Data Structures, Correctness of Algorithm, Analysis of algorithm. Mathematical Analysis of Non-recursive Algorithm - Mathematical Analysis of Recursive Algorithm - Example: Fibonacci Numbers, Sum of 1 to N, etc.	<b>06</b>
<b>2</b>	<b>Growth of Functions &amp; Analysis of Algorithm</b> Introduction - Notion of Algorithm - Fundamentals of Algorithmic Solving , Asymptotic Notation, Bounding Functions, Operations on O-Notations, Bubble Sort, Insertion Sort, Selection Sort, Linear Search, Binary Search, Maximum and minimum	<b>11</b>

<b>3</b>	<b>Recurrences:</b> Introduction, Homogeneous Recurrence, Inhomogeneous Recurrence, Change of variable, substitution methods, Range Transformation, Master's Theorem Recurrence Tree.	<b>07</b>
<b>Section – II</b>		
<b>4</b>	<b>Divide &amp; Conquer</b> Overview of divide and conquer technique, Multiplying large Integers Problem, Problem Solving using divide and conquer algorithm - Binary Search, Sorting (Merge Sort, Quick Sort), Matrix Multiplication, Exponential.	<b>7</b>
<b>5</b>	<b>Greedy Algorithms</b> General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm – Activity selection problem, Elements of Greedy strategy, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Graphs: Shortest paths, The Knapsack Problem, Job Scheduling Problem.	<b>5</b>
<b>6</b>	<b>Dynamic Programming</b> Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming, Calculating the Binomial Coefficient, Making Change Problem, Assembly Line-Scheduling, Knapsack problem, Shortest path, Matrix chain multiplication	<b>8</b>
<b>7</b>	<b>Introduction to NP-Completeness</b> The class P and NP, Polynomial reduction, NP- Completeness Problem, NP-Hard Problems.	<b>4</b>
<b>Instructional Method and Pedagogy</b>		
<ul style="list-style-type: none"> <li>• Activities to be conducted for the topics like analysis.</li> <li>• Feedback by posing a question, quiz, multiple choice questions.</li> <li>• Group work assigning real world application</li> <li>• Power point presentations integrated with video lectures.</li> <li>• Simulators providing a mock scenario</li> <li>• Will use Active Learning methodologies consists of problem-solving exercises, informal small groups, simulations, case studies, role playing, and other activities.</li> <li>• Instructional strategies that engage students in the learning process stimulate critical thinking and a greater awareness of other perspectives.</li> </ul>		

### Reference Books

- Thomas H. Cormen, Charles E. Leiserson, R.L. Rivest, Introduction to algorithms, Prentice Hall of India Publications, New-Delhi.
- Ellis Horowitz, SartajSahni and SanguthevarRajasekaran, Computer Algorithms/ C++, Second Edition, Universities Press, 2007. (For Units II to V)
- Gilles Brassard, Paul Bratley Fundamentals of Algorithms”, 1996, PHI.
- Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, The Design and Analysis of Computer Algorithms, Pearson Education (Singapore) Pvt. Ltd, New Delhi.

### Additional Resources

- NPTEL Lecture Series on Data structure and Algorithm in Computer Science and Engineering, By Naveen Garg, IIT Delhi.

**LAB MANUAL**

List of Tutorials	
Tutorial No	Description
0	<b>Study</b> of vi editor & Prepare list of commands used in VI editor. How to create, compile and execute C program on Linux.
1	- <b>Write</b> a C program to study time measurement functions. - <b>Write</b> C program to create array of size 100. Also fill the random values and print it. - <b>Write</b> C user defined function "CreateList" to create list of N random numbers.
2.	<b>Write</b> UDF & measure time of the following processes: i. Sum of 1 to N numbers ii. Find first N Fibonacci numbers iii. Find primes for a given range
3.	<b>Write</b> a C program to search a value in a list. Also measure the time for best case, worst case and average case using following searching methods: i. Linear Search ii. Binary Search
4.	<b>Implement</b> maximum and minimum algorithms and do performance analysis.
5.	<b>Write</b> C program to compare sorting algorithms i.e. Bubble sort, Insertion sort and Selection sort with their performance. (consider best, worst and average case)
6.	<b>Implement</b> the Merge Sort Algorithm using Divide and Conquer strategy.
7.	<b>Implement</b> the Quick Sort Algorithm using Divide and Conquer strategy.
8.	<b>Write</b> C program to implement the Knapsack algorithm.
9.	<b>Write</b> C program to implement Longest Common Subsequence.
10.	<b>Write</b> C program to prepare Binomial coefficient table.