

Year: M. Tech. I (Semester – I)

Subject Name: Advanced Data Structures and Algorithms

Subject Code: MTCO13103

Type of course: Professional Core Course

Prerequisite (if any): Data structure, Algorithms

List of Courses where this course will be prerequisite: --

Rationale: Data structures play a central role in modern computer science. Data structures are essential building blocks in obtaining efficient algorithms. This course covers a systematic study of the methods of structuring and manipulating data in computing and major advanced data structures which can be used for efficient searching.

Teaching and Examination Scheme:

Teaching Scheme				Theory Marks			Practical Marks		Total
L	T	P	C	TEE	CA1	CA2	TEP	CA3	
4	0	2	5	60	25	15	30	20	150

CA1: Continuous Assessment (assignments/projects/open book tests/closed book tests CA2: Sincerity in attending classes/class tests/ timely submissions of assignments/self-learning attitude/solving advanced problems TEE: Term End Examination TEP: Term End Practical Exam (Performance and viva on practical skills learned in course) CA3: Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course

Content:

Sr. No.	Content	Total Hrs
1	Skip Lists : Need for Randomizing Data Structures and Algorithms, Search and Update Operations on Skip Lists	6
2	Balanced Trees: Binary Search Trees, AVL Trees, Red Black Trees, 2-3 Trees, B-Trees, Splay Trees	10
3	Text Processing and Compression : Sting Operations, Brute-Force Pattern Matching, The Boyer- Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Standard Tries, Compressed Tries, Suffix Tries, The Huffman Coding Algorithm, The Longest Common Subsequence Problem (LCS), Applying Dynamic Programming to the LCS Problem.	7
4	Computational Geometry: One Dimensional Range Searching, Two Dimensional Range Searching, Constructing a Priority Search Tree,	7





	Searching a Priority Search Tree, Priority Range Trees, Quadrees, k-D Trees.	
5	Flow-Networks: Maxflow-mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm. Number Theoretic Algorithms : Modular arithmetic, Chinese remainder theorem, powers of an element, its usage in RSA public-key cryptosystem	10
6	Matrix Operations : 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. Linear Programming : Geometry of the feasibility region and Simplex algorithm	10
7	NP - Completeness : Polynomial time, polynomial time verification, NP completeness and reducibility, NP completeness proofs, NP complete problems Approximation Algorithms : The vertex cover problem, the travelling salesman problem, the set covering problem, set partitioning problem	10

Reference Books:

Sr No.	Title of book /article	Author(s)	Publisher and details like ISBN	Year of publication	Publication Edition
1	Algorithm Design	M T Goodrich, Roberto Tamassia,	John Wiley		
2	Data structures and algorithms	Aho, Hopcroft and Ullman			
3	Introduction to Algorithms	Thomas Cormen, et. al	PHI	2009	2nd Edition



Course Outcomes:

Sr. No.	CO statement	Marks % weightage
CO-1	Develop and analyze algorithms for dynamic data structures and efficient search trees	26%
CO-2	Develop algorithms for text processing applications	12%
CO-3	Identify suitable data structures and develop algorithms for computational geometry.	12%
CO-4	Formulate a real-life problem into a mathematical model and solve it using linear programming.	17%
CO-5	Apply the maximum flow algorithms to solve maximum flow problems	17%
CO-6	Differentiate polynomial and non-polynomial problems.	16%

List of Open learning website:

1. NPTEL course on Data Structures And Algorithms - Video course, Prof. Naveen Garg, Department of Computer Science and Engineering, IIT Delhi
2. Coursera course on Advanced algorithms and Complexity:
<https://www.coursera.org/learn/advanced-algorithms-and-complexity>
3. Flow Networks: <https://web.stanford.edu/class/cs97si/08-network-flow-problems.pdf>,
https://www.coursera.org/lecture/discrete-mathematics/flow-networks-flows-cuts-basic-notions-and-examples6ezT?utm_source=link&utm_medium=page_share&utm_content=vp&utm_campaign=top_button

List of Open Source Software: Linux, CentOS

FOR LAB SESSIONS:

List of Experiments:

Sr No.	Practical
1	Write a menu driven program to implement various operations on the singly linked list (as discussed in the theory class).

2	<p>Write a program for binary search tree to demonstrate following functionalities :</p> <ol style="list-style-type: none"> 1) Create/insert tree, 2) Tree traversals inorder, preorder and post order, 3) Delete operation 4) Copy Tree <p>Above program should also be done using every node as a record of a student and the arrangement of the node should be based on the roll number.</p>
3	Write a program which creates Skip Lists. Implement Insert, Search and Update Operations in Skip-Lists.
4	Write a program which creates AVL Tree. Implement Insert and Delete Operations in AVL Tree. Note that each time the tree must be balanced.
5	Implement 2-3 Tree with basic operations.
6	Implement B Tree with basic operations.
7	Implement Red-Black Tree with basic operations.
8	Implement One Dimensional Range Searching in any language.
9	<p>Text Processing</p> <ol style="list-style-type: none"> 1) Implement a program for String Matching using Boyer-Moore Algorithm on a text file content. 2) Implement a program for String Matching using Knuth-Morris-Pratt Algorithm on a text file content. 3) Implement Huffman-Coding Method. Show the result with a suitable example.
10	Implement Ford-Fulkerson Algorithm.
11	Implement Edmonds-Karp Algorithm.
12	<p>Implement and analyze time complexity of different matrix operations:</p> <ul style="list-style-type: none"> • Matrix Addition/Subtraction • Matrix Multiplication • Matrix Inversion

	<ul style="list-style-type: none">• Transpose of Matrix
13	<p>Solve the following problem of linear programming using pulp library.</p> <p>MINIMIZE: $3X + 5Y$</p> <p>SUBJECT TO:</p> <p>C1: $2X + 3Y \geq 12$</p> <p>C2: $-X + Y \leq 3$</p> <p>C3: $X \geq 4$</p> <p>C4: $Y \leq 3$</p>
14	Implement a set-partitioning problem.

Major Equipment Needed:

