



SARVAJANIK UNIVERSITY
SarvajaniK College of Engineering and
Technology
Bachelor of Technology



Year: B. Tech II (Semester IV)

Subject Name: Design and Analysis of Algorithms

Subject Code: BTCO13403

Type of course: Professional Core Course

Prerequisite (if any): Data Structures, Basic Programming Skills (Python, C, C++, JAVA)

Rationale: This course covers the fundamentals of various algorithm design methodologies. This course teaches students how to analyze and design computer algorithms. Students will be able to examine the asymptotic performance of algorithms after completing this course. Students will be able to use essential algorithmic design paradigms and analytic tools to evaluate the performance of the algorithms after completing the course. This gained knowledge will help students to write algorithms to solve real time problems.

Teaching and Examination Scheme:

Teaching Scheme				Theory Marks			Practical Marks		Total
L	T	P	C	TEE	CA1	CA2	TEP	CA3	
3	0	2	4	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.	Contents	Total Hours
1.	Introduction to Algorithm and Analysis Algorithm and its characteristics, The efficient algorithm, Average, Best and worst case analysis, Asymptotic Notations, Analyzing control statement, Loop invariant and the correctness of the algorithm. Sorting algorithms and their analysis: Bubble sort, Selection sort, Insertion sort, Heap sort, Bucket sort, Radix sort and Counting sort.	08
2.	Divide and Conquer Technique Introduction, Recurrence, Recurrence solving methods - Substitution method, Recursion-tree method and Master method, Problem solving using divide and conquer technique - Binary Search, Merge Sort, Quick Sort, Multiplying large Integers, Strassen's Matrix Multiplication.	06
3.	Greedy Algorithms	06





SARVAJANIK UNIVERSITY
Sarvajanik College of Engineering and
Technology
Bachelor of Technology



	Characteristics of greedy algorithms, Control abstraction, Problem solving using greedy algorithm - The Knapsack problem, Making change problem, Minimum spanning trees using Kruskal's and Prim's algorithms, Single source shortest path problem, Job scheduling problem with deadlines, Huffman Codes.	
4.	Dynamic Programming The Principle of Optimality, Problem solving using Dynamic Programming – Knapsack problem, Making change Problem, All pairs shortest path problem, Matrix chain multiplication, Longest Common sub-sequence.	07
5.	Exploring Graph Algorithms Representation of graphs, Traversing graphs - Breadth-First search and Depth-First search, Topological sorting, Articulation points, Bi-connected components, Strongly connected components.	06
6.	Backtracking and Branch & Bound Backtracking: Knapsack problem, The n-queen problem. Branch & Bound: Knapsack problem, The assignment problem, Traveling Salesman problem.	05
7.	String Matching The naive string matching algorithm, The Rabin-Karp algorithm, The Knuth-Morris-Pratt algorithm.	04
8.	Introduction to NP-Completeness The class P and NP, Polynomial reduction, NP- completeness Problem, NP-hard problems.	03

Suggested Specification table with Marks (Theory): (For B.Tech only)

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
5	15	25	10	5	0

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (Revised Bloom's Taxonomy)

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

Sr No	Title of book /article	Author(s)	Publisher and details like ISBN	Year of publication	Publication Edition
1	Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson,	PHI (ISBN: 978-81-203-2142-0)	1990	4th Edition





SARVAJANIK UNIVERSITY
SarvajaniK College of Engineering and
Technology
Bachelor of Technology



		Ronald L. Rivest and Clifford Stein			
2.	Fundamentals of Algorithmics	Gills Brassard, Paul Bratley	Pearson	1996	1st Edition
3.	Fundamentals of Computer Algorithms	Ellis Horowitz, Sartaj Sahni	Pearson	1984	1st Edition

Course Outcomes:

Sr.No	CO statement	Marks % weightage
CO-1	Compare various algorithmic design techniques and determine an appropriate technique to solve the computational problem in a team or as an individual	15%
CO-2	Utilize algorithmic design technique and mathematical concepts for design and analysis of algorithms	25%
CO-3	Test designed solutions considering different size and characteristics of input data to derive valid conclusions	20%
CO-4	Interpret various pattern matching algorithms and examine them to find a particular pattern in real time problems	20%
CO-5	Understand and analyze major graph algorithms for modeling engineering problems .	15%
CO-6	Differentiate polynomial and non-polynomial problems	5%

List of Open learning website:

- <https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs27/>
- <https://www.coursera.org/specializations/algorithms>
- <https://www.coursera.org/learn/analysis-of-algorithms>

For Lab Sessions:

List of Experiments:

Sr. No.	Practical
1.	Implementation (C Program) of factorial program using: (a) iterative and (b) recursive method





SARVAJANIK UNIVERSITY
SarvajaniK College of Engineering and
Technology
Bachelor of Technology



2.	Implementation and Time analysis of sorting algorithms. Bubble sort, Selection sort, Insertion sort.
3.	Implement linear Counting Sort, Bucket sort, Radix sort algorithm and perform its time analysis.
4.	Implement Merge sort and Quick sort using Divide-and-Conquer and analyze its time complexity.
5.	Implement Heap sort algorithm and calculate its complexity.
6.	Implement Fractional Knapsack Problem using Greedy technique.
7.	Implementation and Time analysis of linear and binary search algorithm.
8.	Implement Making Change Problem using Dynamic Programming.
9.	Implementation and Time analysis of 0/1 Knapsack Problem using Dynamic Programming.
10.	Implement Longest Common Sub-sequence problem using dynamic programming.
11.	Implement the Chained Matrix Multiplication using dynamic programming.
12.	Implementation of Graph Searching Algorithms: DFS and BFS.
13.	Implementation of String Matching using Rabin-Karp Algorithm.
14.	Implement N Queen problem using backtracking.