Kadi Sarva Vishwavidyalaya

Faculty of Engineering & Technology

Third Year Bachelor of Engineering (Computer Engineering)

(In Effect From Academic Year 2019-20)

Subject Code: CE504-N	**Subject Title:** Design & Analysis of Algorithms
Pre-requisite	Programming (C or C++), Data and file structure

Teaching Scheme (Credits and Hours)

Teaching scheme				**Total Credit**	**Evaluation Scheme**					
L	**T**	**P**	**Total**		**Theory**		**Mid Sem Exam**	**CIA**	**Pract.**	**Total**
Hrs	**Hrs**	**Hrs**	**Hrs**		**Hrs**	**Marks**	**Marks**	**Marks**	**Marks**	**Marks**
03	00	02	05	04	03	70	30	20	30	150

Course Objective:

- To understand fundamentals of computer algorithms and algorithmic design paradigms
- To demonstrate a familiarity with major algorithms and data structures.
- To make the students be able to perform analysis of algorithms using asymptotic and empirical approaches
- To Introduce various designing techniques and methods for algorithms
- To develop ability to analyse the algorithms using time and space complexities

Outline of the Course:

Sr. No	**Title of the Unit**	**Minimum Hour**
1	Basics of Algorithms & Mathematics	3
2	Analysis of Algorithms	8
3	Divide and conquer algorithms	8
4	Greedy algorithms	7
5	Dynamic programming	8
6	Graph Algorithms	6
7	Backtracking and Branch and Bound	5
8	Introduction to Complexity Theory	3

Total hours (Theory):48
Total hours (Lab):32
Total hours : 80

Detailed Syllabus

Sr. No	Topic	Lecture Hours	Weight age(%)
1	**Basics of Algorithms and mathematics** • Algorithm definitions and examples • Mathematics for algorithmic sets • Functions and relations • Combinations • Vectors and matrices • Linear inequalities and linear equations	3	6
2	**Analysis of Algorithms** • Orders of Magnitude (Asymptotic notations) • Growth rates, some common bounds (constant, logarithmic, linear, polynomial, exponential) • Time and space complexity • Average and worst case analysis • Analysing control statements • Sorting Algorithms and analysis: Insertion sort, Radix sort	8	17
3	**Divide and conquer algorithms** • Introduction • Recurrence Relations and methods to solve recurrence(substitution, change of variables, master's method, Recurrence tree) • Sorting (Quick sort) • Matrix multiplication • Binary search	8	17
4	**Greedy algorithms** • General Characteristics of greedy algorithms • Problem solving using Greedy Algorithm- Graphs: Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), 0-1 Knapsack problem, Activity selection problem, Making Change Problem	7	14
5	**Dynamic programming** • Introduction • The Principle of Optimality • Problem Solving using Dynamic Programming- Assembly Line Scheduling, Fractional Knapsack problem, Matrix chain multiplication, shortest path, Longest Common Subsequence	8	17

6	**Graph Algorithms:** • An introduction using graphs and games • Traversing Trees– Preconditioning, Depth First Search (DFS), Undirected Graph, Directed Graph, Breath First Search (BFS), Applications of BFS and DFS	6	13
7	**Backtracking and Branch and Bound** • Backtracking –The Knapsack Problem; The Eight queens problem, General Template • Branch and Bound –The Assignment Problem; The Knapsack Problem, The min-max principle	5	10
8	**Introduction to Complexity Theory** • The class P and NP • Polynomial reduction • NP- Complete Problems • NP-Hard Problems • Travelling Salesman problem	3	6
	Total	48	100

Instructional Method and Pedagogy:

- At the start of course, the course delivery pattern, prerequisite of the subject will be discussed.
- Lectures will be conducted with the aid of multi-media projector, black board, OHP etc.
- Attendance is compulsory in lecture and laboratory which carries 10 marks in overall evaluation.
- One internal exam will be conducted as a part of internal theory evaluation.
- Assignments based on the course content will be given to the students for each unit and will be evaluated at regular interval evaluation.
- Surprise tests/Quizzes/Seminar/tutorial will be conducted having a share of five marks in the overall internal evaluation.
- The course includes a laboratory, where students have an opportunity to build an appreciation for the concepts being taught in lectures.
- Experiments shall be performed in the laboratory related to course contents.

Learning Outcome:

On successful completion of the course, the student will:

- Be able to check the correctness of algorithms using inductive proofs and loop invariants.
- Be able to compare functions using asymptotic analysis and describe the relative merits of worst-, average-, and best-case analysis.
- Be able to solve recurrences using the master, the iteration, and the substitution method.
- Become familiar with a variety of sorting algorithms and their performance characteristics (eg, running time, stability, space usage) and be able to choose the best one under a variety of requirements.

- Be able to understand and identify the performance characteristics of fundamental algorithms and data structures and be able to trace their operations for problems such as sorting, searching, selection, operations on numbers, polynomials and matrices, and graphs.
- Be able to use the design techniques introduced i.e. dynamic programming, greedy algorithm etc. to design algorithms for more complex problems and analyze their performance.
- Find optimal solution by applying various methods.
- Become familiar with the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.
- Differentiate polynomial and non polynomial problems.

Reference Books:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E .Leiserson, Ronald L. Rivestand Clifford Stein, PHI
2. Fundamental of Algorithms by Gills Brassard, Paul Bratley, PHI.
3. Design and Analysis of Computer Algorithms by Aho, Hopcroft and Ullman, Pearson
4. The Algorithm Design Manual By Steve s. Skiena

E-Resource : https://nptel.ac.in/courses/106106131/
https://onlinecourses.nptel.ac.in/noc18_cs37

List of experiments

No	Name of Experiment
1	**Basics:** Find out Big - Oh and Big–Omega of the function. Take necessary data like degree of the function, coefficients, etc.
2	**Revision of Data Structures**: Write a program to implement: a. A Queue b. A Stack c. A Queue using two Stacks d. A Stack using two Queues
3	**Some Basic Algorithms:** Write an algorithm and find the efficiency of the same for following problems: a. Finding Factorial – Iterative Approach b. Finding Factorial – Recursive Approach c. Printing Fibonacci Series – Iterative Approach d. Printing Fibonacci Series – Recursive Approach

4	**Basic Sorting and Searching Techniques:** Design an algorithm and implement a program for: a. Insertion Sort b. Selection Sort c. Linear Search d. Radix Sort
5	**Divide and Conquer Approach:** Design an algorithm and implement a program for: a. Quick Sort b. Binary Search
6	**Greedy Approach:** Design an algorithm and implement a program to solve: a. Making Change Problem b. Knapsack Problem
7	**Dynamic Programming:** Design an algorithm and implement a program to solve: a. Knapsack Problem b. Longest Common Subsequence Problem c. Finding Optimal Matrix Chain Order Problem
8	**Graph Algorithms:** Design an algorithm and write a program to implement: a. Depth First Search of a graph b. Breadth First Search of a graph
9	**Graph Algorithms:** Design an algorithm and implement a program for: a. Kruskal’s method of finding Minimum Spanning Tree b. Prim’s method of finding Minimum Spanning Tree
10	Design an algorithm and implement a program for The Assignment Problem in Branch and Bound