Kadi Sarva Vishwavidyalaya

Faculty of Engineering & Technology

Third Year Bachelor of Engineering (CE/IT)

(In Effect From Academic Year 2019-20)

Subject Code: CT604D-N	**Subject Title:** Advanced Algorithms
Pre-requisite	Programming (C or C++), Data structure, Analysis of Algorithms, Set Theory

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:

- The complexity and performance analysis of different algorithms using asymptotic and empirical approaches
- Proper selection of the algorithms best suited for given applications
- To introduce various designing techniques and methods for algorithms
- To be familiar with various data structures and advanced data structures
- Applying efficient algorithms in engineering problems

Outline of the Course:

Sr. No	Title of the Unit	Minimum Hours
1	Introduction	10
2	Data Structures and Advanced Data Structures	10
3	Advanced Computer Algorithms	8
4	Advanced Design and Analysis Techniques	6
5	Graph Algorithms	6
6	Backtracking and Branch and Bound	3
7	String Matching	3
8	Introduction to NP-Completeness	2

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

Kadi Sarva Vishwavidyalaya

Faculty of Engineering & Technology

Third Year Bachelor of Engineering (CE/IT)

(In Effect From Academic Year 2019-20)

Detailed Syllabus

Sr. No	Topic	Lecture Hours	Weight Age(%)
1	**Introduction:-**Algorithms, Analyzing algorithms, Growth of Functions-Asymptotic notation, Mathematical Background for algorithm analysis, Sorting algorithms and Analysis (Compare-exchange, divide-conquer, linear time, tree sorting), Applications of sorting and searching	10	21
2	**Data Structures:-**Introduction of basic data structures like stack, queue, linked-list, binary tree, binary search tree, AVL tree,Red-Black tree, splay tree, Augmenting data structures **Advanced Data Structures:-**B-trees, Binomial heaps, Fibonacci heaps	10	21
3	**Advanced Computer Algorithms:-**Network flow problems(max flow and min-cut), Sorting networks (odd-even merge, bitonic, butterfly), Parallel algorithms (Convex-hull, sorting, searching, merging), Approximation algorithms (vertex cover, TSP, set covering, subset sum)	8	16
4	**Advanced Design and Analysis Techniques:-** **Dynamic Programming:-**Elements of dynamic programming, Matrix-chain multiplication **Greedy Algorithms:**-Elements of the greedy strategy, Huffman codes, **Amortized Analysis:-**Aggregate analysis, The accounting method, The potential method	6	13
5	**Graph Algorithms:-** **Single-Source Shortest Paths:-**The Bellman-Ford algorithm, Dijkstra's algorithm, Difference constraints and shortest paths **All-Pairs Shortest Paths:-**The Floyd-Warshall algorithm **Maximum Flow:-**Flow networks, The Ford-Fulkerson method	6	13
6	**Backtracking and Branch and Bound:-** Introduction, The Eight queens problem , Knapsack problem, Travelling Salesman problem, Minimax principle	3	6
7	**String Matching:-** Introduction, The naive string matching algorithm, The Rabin-Karp algorithm, String Matching with finite automata, The Knuth-Morris-Pratt algorithm. Recovery,The Boyer-Moore Algorithms.	3	6
8	**Introduction to NP-Completeness:-** The class P and NP, Polynomial reduction, NP- Completeness Problem, NP-Hard Problems. Travelling Salesman problem, Hamiltonian problem, Approximation algorithms	2	4
	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:-

- After learning the course the students should be able to:
- Analyse the asymptotic performance of algorithms.
- Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
- Find optimal solution by applying various methods.
- Apply pattern matching algorithms to find particular pattern.
- Differentiate polynomial and non-polynomial problems.
- 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/recursion tree, 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.
- Learn the graph algorithms and their analysis. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyse them.

- Be able to use the design techniques introduced i.e. dynamic programming, greedy algorithm etc. to design algorithms for more complex problems and analyse their performance.

Reference Books:-

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, PHI.
2. The Design and Analysis of Computer Algorithms byAho, Hopcroft, Ullman.
3. Algorithms Design and Analysis by Udit Agrawal, Dhanpat Rai & co.
4. Design Analysis And Algorithms by Hari Mohan Pandey
5. The Algorithm Design Manual By Steve s. Skiena
6. Fundamental of Algorithms- Theory and Practice by Gilles Brassard and Paul Bratley
7. Randomized Algorithms by Rajeev Motwani and PrabhakarRaghavan
8. Research papers and NPTEL videos

List of experiments

No	Name of Experiment
1	Implement insertion sort, bubble sort and selection sort along with the cost analysis and frequency counts.
2	Implement merge sort and quick sort along with the cost analysis and frequency counts.
3	Implement heap sort, radix sort and count sort along with the cost analysis and frequency counts
4	Implement AVL Tree.
5	Implement B Tree.
6	Implement any one algorithm with Dynamic Programming.
7	Implement any one algorithm with Greedy Programming.
8	Implement Minimum Spanning Tree.
9	Implement any one approximation algorithm.
10	Implement any one randomized algorithm.
11	Implementation of The Eight queens problem.
12	Implement any one string matching algorithm.