SILVER OAK UNIVERSITY

College of Technology (01)

Degree Engineering Course (Computer Engineering)

Subject Name: Analysis and Design of Algorithms

Subject Code: 1010043316

Semester: 5th

Prerequisite:

1. **Programming (C or C++)**
2. **Data and file structure**

Objectives:

1. Analyze the asymptotic performance of algorithms.
2. Demonstrate a familiarity with major algorithm design techniques.
3. Apply important algorithmic design paradigms and methods of analysis.
4. Solve simple to moderately difficult algorithmic problems arising in applications.
5. Able to demonstrate the hardness of simple NP-complete problems.

Teaching and Examination Scheme:

Teaching Scheme					Evaluation Scheme				Total Marks
L	T	P	Contact Hours	Credits	Theory		Practical		
					CIE (TH)	ESE (TH)	CIE (PR)	ESE (PR)	
4	0	2	6	5	40	60	20	30	150

Content:

Unit No.	Course Contents	Teaching Hours	Weightage %
1	**Basics of Algorithms and Mathematics:** What is an algorithm?, Properties of Algorithm, Time and Space Complexity, detailed analysis of algorithm, Mathematics for Algorithmic Sets, Functions and Relations, Vectors and Matrices, Linear Inequalities and Linear Equations.	4	3

2	**Analysis of Algorithm:** The efficient algorithm, Average, Best and worst case analysis, Amortized analysis , Asymptotic Notations(Big-O, Big-Ω and Big-Θ Notations their Geometrical Interpretation and Examples.), Recurrences: Recursive Algorithms and Recurrence Relations, Solving Recurrences, Analyzing control statement, Loop invariant and the correctness of the algorithm, Sorting Algorithms and analysis: Bubble sort, Selection sort, Insertion sort, Shell sort Heap sort, Sorting in linear time : Bucket sort, Radix sort and Counting sort	**10**	**20**
3	**Divide and Conquer Algorithm:** Introduction, Recurrence and different methods to solve recurrence, Multiplying large Integers Problem, Problem Solving using divide and conquer algorithm - Binary Search, Max-Min problem, Sorting (Merge Sort, Quick Sort), Matrix Multiplication, Exponential.	**8**	**15**
4	**Dynamic Programming:** Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming – Calculating the Binomial Coefficient, Making Change Problem, Assembly Line-Scheduling, Knapsack problem, All Points Shortest path, Matrix chain multiplication, Longest Common Subsequence.	**7**	**15**
5	**Greedy Algorithm** 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, Huffman code.	**7**	**15**
6	**Exploring Graphs:** An introduction using graphs and games, Undirected Graph, Directed Graph, Traversing Graphs, Depth First Search, Breath First Search, Topological sort, Connected components	**6**	**10**
7	**Backtracking and Branch and Bound:** Introduction, The Eight queens problem , Knapsack problem, Travelling Salesman problem, Minimax principle	**4**	**10**
8	**String Matching:**	**4**	**5**

	Introduction, The naive string matching algorithm, The Rabin-Karp algorithm, String Matching with finite automata, The Knuth-Morris-Pratt algorithm.		
9	**Introduction to NP-Completeness:** The class P and NP, Polynomial reduction, NP- Completeness Problem, NP-Hard Problems. Travelling Salesman problem, Hamiltonian problem, Approximation algorithms, Randomized algorithms, Class of problems beyond NP – P SPACE	**5**	**7**

Course Outcome:

Sr. No.	CO statement	Unit No	Weightage %
CO-1	Analyze the asymptotic performance of algorithms.	**1,2**	**15%**
CO-2	Derive and solve recurrences describing the performance of divide-and-conquer algorithms.	**3**	**15%**
CO-3	Find optimal solution by applying various methods.	**4,5**	**30%**
CO-4	Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.	**6,7**	**30%**
CO-5	Apply pattern matching algorithms to find particular pattern.	**8**	**5%**
CO-6	Differentiate polynomial and nonpolynomial problems.	**9**	**5%**

- **Teaching and Learning Mode:**

1. The course includes a laboratory, where students get the opportunity to practically apply the theoretical knowledge they have acquired in the lectures.
2. Lectures with attractive PowerPoint presentations
3. Different experiments shall be carried out during the practical sessions.

- **List of Experiments/Tutorials:**
    1. Implementation and Time analysis of sorting algorithms.

Bubble sort, Selection sort, Insertion sort, Merge sort and Quicksort

2. Implementation and Time analysis of linear and binary search algorithm.
3. Implementation of max-heap sort algorithm
4. Implementation and Time analysis of factorial program using iterative and recursive method
5. Implementation of a knapsack problem using dynamic programming.
6. Implementation of chain matrix multiplication using dynamic programming.
7. Implementation of making a change problem using dynamic programming
8. Implementation of a knapsack problem using greedy algorithm
9. Implementation of Graph and Searching (DFS and BFS).
10. Implement prim's algorithm
11. Implement kruskal's algorithm.
12. Implement LCS problem.

- **Major Equipment/ Instrument(Software/Hardware):**
  Computers with latest software configuration, Turbo C

- **Books Recommended:**

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivestand Clifford Stein, PHI.
2. Fundamentals of Algorithms – E. Horowitz et al.
3. Fundamental of Algorithms by Gills Brassard, Paul Bratley, PHI.
4. Introduction to Design and Analysis of Algorithms, Anany Levitin, Pearson.
5. Foundations of Algorithms, Shailesh R Sathe, Penram
6. Design and Analysis of Algorithms, Dave and Dave, Pearson.

- **List of Open Source Software/learning website:**

1. NPTEL tutorials
2. http://www.coursera.org/