Design and Analysis of Algorithms

Course Title: Design and Analysis of Algorithms
Course No: CSC314
Nature of the Course: Theory + Lab
Semester: V

Full Marks: 60 + 20 + 20
Pass Marks: 24 + 8 + 8
Credit Hrs: 3

Course Description: This course covers the basic concepts of computers and information technology including introduction, hardware, software, memory, input/output, data representation, database, networks and data communication, Internet, multimedia, and computer security.

Course Description: This course introduces basic elements of the design and analysis of computer algorithms. Topics include asymptotic notations and analysis, divide and conquer strategy, greedy methods, dynamic programming, basic graph algorithms, NP-completeness, and approximation algorithms. For each topic, beside in-depth coverage, one or more representative problems and their algorithms shall be discussed.

Course Objectives:

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

Detail Syllabus:

Unit 1	**Foundations of Algorithm Analysis**	**Teaching Hours (4)**
Algorithms and its properties	Definition of algorithms and brief explanation about the basic properties of algorithms	1 hr
RAM model	Explanation of the RAM model and its use for algorithm analysis.	
Time and Space Complexity	Concepts of Time and Space Complexity with best case, worst case , average case	
Detailed Analysis of algorithms	Detailed Analysis with examples like factorial of an integer using RAM model.	
Concept of Aggregate Analysis	Definition, brief explanation of Aggregate Analysis with example.	
Asymptotic Notations:	Concept, definition of Asymptotic notation: Big-O, Big-Ω and Big-Θ Notations and their Geometrical Interpretation and Examples.	1 hr
Recursive Algorithms	Brief overview of recursion and , Recursive Algorithms	2 hrs
Recurrence Relations	Definitions of Recurrence Relations with example. Uses of Recurrence Relations in Algorithm Analysis.	
Solving Recurrences	Recursion Tree Method, Substitution Method, Application of Masters Theorem for solving recurrence relations. Examples	

Unit 2	**Iterative Algorithms**	**Teaching Hours (4)**
Basic Algorithms	Algorithm for GCD, Fibonacci Numbers and analysis of their time and space complexity.	1 hr
Searching Algorithms:	Sequential Search and its analysis	3 hrs
Sorting Algorithms:	Description of Bubble Sort, Selection Sort and Insertion Sort with their complexity analysis.	
Unit 3	**Divide and Conquer Algorithms**	**Teaching Hours (8)**
Concepts	Concept and applications of divide and conquer approach in algorithm design.	2 hr
Searching Algorithms:	Concept and detail description of Binary Search algorithms and its analysis, Finding Minimum and maximum element in a list of items(Min-Max algorithm) and their analysis.	
Sorting Algorithms:	Merge Sort algorithm, examples and its time and space complexity	1 hrs
	Concepts of partitioning, Quick Sort algorithm and its analysis (Best Case, Worst Case and Average Case). Examples, Randomized Quick Sort and its analysis.	2 hrs
	Concept of Heap Data Structures(max , min). Heap Sort Algorithm (with Build Heap and Heapify) and its complexity analysis.	1 hr
Order Statistics	Concepts of Order statistics, Median order. Brute-force approach for selection	1hrs
	Selection in Expected Linear Time and its analysis.	
	Selection in Worst Case Linear Time algorithm and its complexity analysis.	1 hr
Unit 4	**Greedy Algorithms**	**Teaching Hours (6)**
Introduction to Greedy Approach	Concept of Optimization Problems and Optimal solution. Introduction of Greedy Strategy for algorithm design. Elements of Greedy Strategy(Greedy Choice Property, Optimal Substructure Property)	1 hr
Greed Algorithms:	Concept of Knapsak problem, Algorithm for Fractional Knapsack Problem examples and analysis of its complexity.	**1 hr**
	Concept of Job Sequencing Problem with deadline. Algorithm for Job Sequencing with deadline and its time complexity.	**1 hr**
	Kruskal's and Prim's algorithms for Minimum Spanning Tree, their examples and complexity analysis. Correctness .Dijkastra Shortest Path Algorithms , example and its time complexity.	**2 hr**
Huffman Coding:	Purpose of Huffman Coding, Prefix Codes, Huffman Tree, Huffman Coding Algorithm, example and its Analysis.	1hr

Unit 5	**Dynamic Programming**	**Teaching Hours (8)**
Introduction	Concepts of Dynamic Programming approach for algorithm design, Greedy Algorithm vs Dynamic Programming, Recursion vs Dynamic Programming. Elements of Dynamic Programming Approach	1.5hrs
D P Algorithms:	Concept of Matrix Chain Multiplication, its Algorithm ,examples and complexity analysis	1.5 hrs
	String Editing Algorithm(edit distance problem with insertion, deletion, replace operation) and its complexity analysis	1 hr
	0-1 Knapsack problem and its complexity analysis.	3hr
	Floyd Warshall Algorithms for all pair shortest path problem, example and its complexity analysis.	
	Travelling Salesman Problem and its analysis	
Memoization Strategy	Concept of Memoization. Dynamic Programming vsMemoization.	1hr
Unit 6	**Backtracking**	**Teaching Hours (5)**
Introduction	Concept of Backtracking Approach. Recursion vs Backtracking	1hr
Backtracking Algorithms	Concept of Subset Sum, Algorithm for Subset-Sum, its example and Complexity Analysis.	**4 hrs**
	Zero-One Knapsack Problem, algorithm with backtracking approach and its analysis.	
	N-Queen Problem and their Analysis	
Unit 7	**Number Theoretic Algorithms**	**Teaching Hours (5)**
Introduction	Concept of Number Theoretic Notation.	2 hrs
	Concept of Modular Linear Equations. Chinese Remainder Theorem.	
Solving Modular Linear Equations	Euclid's and Extended Euclid's Algorithms for solving Modular Linear Equations.	2 hrs
Primility Testing	Miller-Rabin Randomized Primility Test and Analysis	1hr
Unit 8	**NP Completeness**	**Teaching Hours (5)**
Tractable and Intractable Problems, Complexity Classes	Concept of tractable and intractable problems, Polynomial Time and Super Polynomial Time complexity.	2 hr
	P , NP , NP Complete, NP Hard with Examples	
NP Complete Problems	NP Completeness and Problem Reducibility, Concept of Cooks Theorem(Without Proof). Proof of NP Completeness(CNF-SAT, Vertex Cover and Subset-Sum Problem)	**2 hrs**
Approximation Algorithms	Concept and Application, Vertex Cover Problem, Subset Sum Problem	**1hr**

Laboratory Works:

This course can be learnt in effective way only if we give focus is given in practical aspects of algorithms and techniques discussed in class. Therefore student should be able to implement the algorithms and analyze their behavior.

For the laboratory work, students should implement the following algorithms in C/ C++ and perform their analysis for time and space complexity.

1. Basic iterative algorithms GCD algorithm, Fibonacci Sequences, Sequential and Binary Search.
2. Basic iterative sorting algorithms: Bubble Sort, selection Sort, Insertion Sort.
3. Binary Search with Divide and conquer approach.
4. Merge Sort, Heap sort, Quick Sort, Randomized Quick Sort.
5. Selection Problem with divide and Conquer approach
6. Fractional Knapsack Problem, Job sequencing with deadline, Kruskal's algorithm, Prims algorithm, Dijkstra's Algorithm
7. Implement the dynamic programming algorithms.
8. Algorithms using Backtracking approach.
9. Implement approximation Algorithm.

Recommended Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to algorithms", Third Edition.. The MIT Press, 2009.
2. Ellis Horowitz, SartajSahni**,** SanguthevarRajasekiaran, "Computer Algorithms", Second Edition, Silicon Press, 2007.
3. Kleinberg, Jon, and Eva Tardos**, "** Algorithm Design**" ,** Addison-Wesley**,** First Edition, 2005

Tribhuvan University

Institute of Science and Technology

Model Question

Bachelor Level/Third Year / Fifth Semester/Science — Full Marks: 60

Computer Science and Information Technology — Pass Marks: 24

(CSc. 314- Design and Analysis of Algorithm) — Time : 3 hours.

Section A

Attempt any two questions. (2 × 10 = 20)

1. Why do you need the algorithm analysis? Discuss about RAM model for analysis of algorithms. Also discuss about Big Oh, Big Omega and Big theta with examples. (2+3+5)
2. Discuss the order statistics. Explain about the worst case linear time selection algorithm and analyze its time complexity. (2 + 8)
3. Explain in brief about the Dynamic Programming Approach for algorithm design. How it differs with recursion? Explain the Floyd Warshall algorithm to compute the all pair shortest path in graph and analyze its time complexity. (4 + 6)

Section B

Attempt any eight questions. (8 × 5 = 40)

4. Write the algorithm for Binary Search with divide and conquer approach and explain its complexity. (5)
5. Solve the following recurrence relations using master method. (2.5 + 2.5)
   a. $T(n) = 3T(n/2) + n$
   b. $T(n) = 2T(n/4) + \sqrt{n}$
6. What is prefix code? Explain Huffman algorithm to compute the prefix codes. (5)
7. Write the algorithm for insertion sort and explain its time complexity (5)
8. What do you mean by memoization strategy? Compare memoization with dynamic programming (5)
9. Explain backtracking with suitable example. (5)
10. Explain the Euclid's method to solve the modular linear equations with example (5)
11. Explain in brief about the complexity classes P, NP and NP Complete (5)
12. Write short notes on
    a. Tractable and Intractable Problems
    b. Approximation Algorithms

The END