RAMAKRISHNA MISSION VIDYAMANDIRA

CBCS Syllabus B.Sc. Computer Science Honours

Semester-III

Course Code: CMSA CC 5 Credit: 6
Course Type: Core Course

Course Outcome:

i) To learn mathematical background for analysis of algorithm.
ii) To be able to analyze complexity of any algorithm.
iii) To learn various computational problem solving approaches.
iv) To learn advanced tree and graph applications
v) Basic idea of role of computation in biology.
vi) To develop basic idea of complexity class of algorithms.
vii) Solving problems on various algorithm techniques.
viii) Developing practical knowledge on implementation of Genetic algorithm.

CMSA CC 5 T: Design and Analysis of Algorithm

Credit: 4 **Marks: 50**

Introduction to Algorithms: Definition, Correctness of Algorithm, Recursive and Non-recursive algorithms. [2 L]

Asymptotic Complexity Analysis of Algorithms: Space and Time Complexity, Efficiency of an algorithm. Growth of Functions, Polynomial and Exponential Complexity, Asymptotic Notations: Big O Notation, Big Omega and Big-Theta Notations, Properties: Best case/worst case/average case analysis of well-known algorithms, Amortized analysis. [10 L]

Algorithm Design Techniques: Concepts and simple case studies of Greedy algorithms; Divide and conquer: Basic concepts, Case study of selected searching and sorting problems as divide and conquer techniques: Strassen's Matrix Multiplication Method ;Dynamic programming: General issues in Dynamic Programming, Case study of Binomial Coefficient computation; Backtracking: Basic Concepts, Case Study of N Queens Problem; Branch and Bound: Basic Concepts, Case Study of Job Sequencing with Deadline Problem. [23 L]

Graph Representation and Algorithm: Linked Representation, Matrix representations of graphs, Incidence, Adjacency and Circuit matrices, Graph operations; Graph searching algorithms: BFS, DFS, Minimal spanning tree algorithms: Prim's Algorithm, Kruskal's Algorithm, Shortest path algorithms: Bellman Ford Algorithm, Dijkstra's Algorithm, Floyd-Warshall Algorithm. [12 L]

Algorithm in Computational Biology: Role of computation in Biology; Genetic Algorithm: Basic Concepts, Reproduction, Cross over, Mutation, Fitness Value, Optimization; Genome rearrangements by Greedy algorithm; [8 L]

Classification of Problems: P, NP, Cook's Theorem (Statement Only). [5 L]

CMSA CC 5 P: Design and Analysis of Algorithm Laboratory

Credit: 2 **Marks: 25**

Implementing problems on Greedy approach, Divide and conquer, Dynamic programming. [15 L]

Implementation of Graph algorithms: BFS, DFS, Prim's Algorithm, Kruskal's Algorithm, Bellman Ford Algorithm, Dijkstra's Algorithm, Floyd-Warshall Algorithm. [15 L]

Implementation of programs on Genetic Algorithm:

1. Using genetic algorithm, find maximum or minimum value of a given function using the concept of biological chromosomes and genes. Also show crossover, mutation etc in the implementation.

2. Implement the following using genetic algorithm.

Produce a target string starting from a random string of the same length. The target string is given by the user. Consider the followings for the implementation.

- Characters A-Z, a-z, 0-9 and other special symbols are considered as genes
- A string generated by these character is considered as chromosome/solution/Individual
- Fitness score is the number of characters which differ from characters in target string at a particular index. So individual having lower fitness value is preferred.

3. Implement the following using genetic algorithm.

Given a set of 5 genes, each gene can hold one of the binary values 0 and 1.
The fitness value is calculated as the number of 1's present in the genome. If there are five 1s, then it is having maximum fitness. If there are no 1s, then it has the minimum fitness.
Also try to maximize the fitness function to provide a population consisting of the fittest individual, i.e. individuals with five 1s.

4. Using genetic algorithm, find an equation composed of addition/subtraction for auser-given target number.
Example: Input target number: 27, Output:-7+29-8-1 or - 15+28-16 or 2+24-5+6

5. Implement a number or password guessing program using genetic algorithm.

6. Compare two DNA strands of two different organisms using Longest Common Subsequence (LCS) problem to determine the measure of how closely related the two organisms are.

[A strand of DNA consists of a string of molecules called bases, where the possible bases are Adenine (A), Guanine (G), Cytosine (C), and Thymine(T).Representing each of these bases by its initial letter, we can express a strand of DNA as a string over the finite set {A; C; G; T}. Example of the DNA of one organism may beS1 =ACCGGTCGAGTGCGCGGAAGCCGGCCGAA, and the DNA of another organism may be S2 =GTCGTTCGGAATGCCGTTGCTCTGTAAA.] [10 L]

Recommended Books:

1. Introduction to Algorithms by Cormen et al, 3rdEdition, PHI Learning.
2. Algorithm Design by Kleinberg, Tardos; Pearson.
3. An Introduction to Bioinformatics Algorithms by Neil C Jones, Pavel Pevzner; MIT Press.
4. Algorithms & Data structure by Ellis Horowitz, H.Sahani, SanguthevarRajasekaran, 2nd Edition; Universities Press.
5. The Art of Programming (Vol.1 and Vol.2) by Donald. E. Knuth, 3rd Edition, Pearson.
6. Introduction to Soft Computing: Neuro-Fuzzy and Genetic Algorithm by Roy, Chakraborty; 1st Edition; Pearson.
7. Computer Algorithms: Introduction to Design and Analysis by Sara Baase, Van Gelder; 1st Edition; Pearson.