CSE330:COMPETITIVE CODING APPROACHES-TECHNIQUES

L:2 T:0 P:1 Credits:3

Course Outcomes: Through this course students should be able to

CO1 :: analyze the space and time complexity of an algorithms

CO2 :: determine primality testing on different algorithm

CO3 :: apply recursion techniques on various dynamic programming problems

CO4 :: apply tabulation and memorization in dynamic programming problem

CO5 :: apply various sorting techniques by computing O(n log n) complexity

CO6 :: identify the sorting algorithm in real world problem solving

Unit I

Behaviour Analysis : Introduction to limits and behaviour of logic, understanding taxonomy in worst case, analysing the effectiveness and efficiency of algorithms, measuring time and space complexity of algorithm, trade-off concept

Unit II

Primality Testing : Introduction to Primality Testing, O(sqrt(n)) Algorithm for Primality Testing, Factorization of a number, Finding prime factors by taking the square root, Fermat method, Sieve of Eratosthenes, Segmented Sieve, Sieve of Atkins, Mansi and her series, Collections of Pens, next prime palindrome

Unit III

Recursion and Advanced Techniques : Introduction to recursion, base condition, Solving problems using recursion, Classic and Modern Approaches, Direct vs. Indirect Recursion, Tailed vs. Non-Tailed Recursion, Memory Allocation in Recursion, Advantages & disadvantages of recursive programming, Backtracking, Memoization, recursive problems- next happy number, sum string, water overflow

Unit IV

Basic Dynamic Programming : Introduction to Dynamic Programming, Tiling problem, Tabulation vs Memoizatation, Optimal Substructure Property, Overlapping Subproblems Property, Dynamic Programming Process and Techniques, Formulating Dynamic Programming Problems

Unit V

Dynamic Programming Problems : Binomial coefficient, Box Stacking, Integer Knapsack Problem (Duplicate Items Forbidden), Edit Distance, Longest Increasing Subsequence(LIS), Longest Common Subsequence (LCS), Balanced Partition Problem

Unit VI

Efficient Sorting Algorithms & Analysis : Introduction to O(n logn) Sorting Algorithms, Iterative & Recursive Merge Sort, Quick Sort, Sorting Elements by Frequency, Finding Minimum Length Sorted Sub-array to Sort an Array, Sorting Strings, case-specific sorting of strings, Count Distinct Pairs with Difference of K

List of Practicals / Experiments:

List of Practical's

- Finding prime factors by taking the square root
- Fermat method
- Sieve of Eratosthenes
- Segmented Sieve
- Sieve of Atkins
- Tiling problem
- Longest increasing subsequence
- Longest common subsequence
- Binomial coefficient

- Box Stacking
- Integer Knapsack Problem (Duplicate Items Forbidden)
- Edit Distance
- Balanced Partition
- Merge Sort
- Quick Sort
- Counting Sort
- String sorting

Text Books:

1. CRACKING THE CODING INTERVIEW by GAYLE LAAKMANN MCDOWELL, CAREERCUP

References:

1. PROGRAMMING PEARLS by JOE BENTLEY, PEARSON