

CSE329:PRELUDE TO COMPETITIVE CODING

L:0 T:0 P:3 Credits:3

Course Outcomes: Through this course students should be able to

- CO1 :: relate the theoretical knowledge and insights gained to formulate working code
- CO2 :: devise time and space efficient algorithms to solve abundant ubiquitous problems
- CO3 :: identify the intricacies present in the design of a solution to devise an optimal solution
- CO4 :: deduce the appropriate and efficient algorithms and data structures for optimal solution to the problems at hand
- CO5 :: extend and utilize the knowledge base of various algorithmic paradigms to build optimized solutions to real world problems
- CO6 :: validate the logic building and code formulation by designing optimal code capable of passing various test cases

List of Practicals / Experiments:

Basic Data Structures

- Arrays, Declaring and processing 1D and 2D arrays
- Insertion in an array
- Deletion from array
- Array rotations
- Array arrangement, rearrangement
- Matrix multiplication, Lower triangular and upper triangular matrix of array, Different operations on Matrices
- Print a matrix in spiral form, Find distinct elements common to all rows in a matrix
- String declaration and manipulation
- K maximum sum from two arrays
- Missing characters to make a string Pangram, Rearrange characters so that no two adjacent characters are same
- Remove minimum number of characters so that two strings become anagram
- Creation of stack using arrays, Creation of arrays using linked list
- Creation of queue using array, Creation of queue using linked list
- Implement two Stacks in an array
- Implement Stack using Queues
- Design a stack with operations on middle element
- Implementation of Deque using circular array, Circular Queue
- Reversing a Queue

Basic math operations (addition, subtraction, multiplication, division and exponentiation)

- Fast modulo multiplication
- Exponential squaring
- N-th non-square number
- Modular Exponentiation
- Modular multiplicative inverse
- Euler's Totient Function
- Sum of middle row and element in matrix
- Checking if all rows of a matrix are circular rotations of each other

GCD and Primality testing

- Basic Euclidean algorithm
- Extended Euclidean algorithm
- Total number of divisors of a number
- Finding all prime factors of a number
- K-jagged numbers
- Stormer numbers
- Finding the prime factors by taking the square root
- Frugal number
- P-smooth numbers in given ranges
- Lemoine's Conjecture
- Problems based on GCD and primality testing

Greedy techniques

- Greedy problem solving paradigm
- Locally optimal choice
- Global optimal choice
- Job Sequencing problem
- Job Selection problem
- Minimum product subset of an array
- Maximum product subset of an array
- Minimum sum of product of two arrays
- Bin packing problem
- Activity Selection problem
- Fractional Knapsack problem
- Connect n ropes with minimum cost
- Coin change problem
- Problems based on greedy techniques

Naive Pattern Search

- The ubiquitous naïve pattern search problem
- KMP algorithm
- Rabin-Karp algorithm
- Introduction to Suffix array
- Checking if two strings are rotations of each other
- Largest connected component on a grid
- Check if a string is substring of another string
- Longest prefix which is also a suffix
- Problems based on naive pattern search

Searching techniques

- Iterative and recursive binary search
- Sublist search
- Find the missing number
- Search an element in a sorted and rotated array
- Recursive function to perform substring search
- Find the K most frequent words from a string
- Find a pair with a given difference

- Find a peak element
- Problems based on searching techniques

References:

1. CRACKING THE CODING INTERVIEW by GAYLE LAAKMANN MCDOWELL, CAREERCUP
2. DATA STRUCTURES AND ALGORITHMS : CONCEPTS, TECHNIQUES AND APPLICATIONS by G. A. V. PAI, MCGRAW HILL EDUCATION