

CSE331:CODING PEARLS

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

Course Outcomes: Through this course students should be able to

CO1 :: relate the theoretical as well as practical knowledge to form an amalgamation of working code

CO2 :: employ a working combination of solution to ubiquitous problems which are time and space efficient

CO3 :: identify and comprehend the inner workings behind the design of an optimal solution

CO4 :: illustrate the usage of algorithms and data structures in the design of an optimal solution towards a problem

CO5 :: recall the knowledge obtained from various algorithmic paradigms to formulate optimal solutions to real world problems

CO6 :: examine and utilise knowledge to build and design reliable code which is capable of passing various test cases

Unit I

Heaps (Priority Queues) : Max Heap, Min Heap, Heap Sort, K'th Largest element in array, Sort an almost sorted array, Connect n ropes with minimum cost, Array representation of binary heap

Unit II

Disjoint Set Union : Union-Find algorithm, Union by Rank and Path Compression

Unit III

Binary trees : Types of Binary Tree, Insertion in a tree, Deletion in a tree, Tree traversals, Inorder, Preorder, Postorder, Inorder traversal without recursion, Print Postorder traversal from given Inorder and Preorder traversals, Level Order Tree Traversal

Unit IV

Problems based on Binary Trees : Populate Inorder Successor for all nodes, Find n-th node of inorder traversal, Level Order traversal in spiral form, Boundary traversal of binary tree, Finding Lowest Common Ancestors, Sum of nodes in a binary tree having only the left child nodes, Diameter of a tree, Determine if Binary Tree is height balanced, Convert Binary Search Tree to Min Heap

Unit V

Graph algorithms : Graph and its representations, Breadth First Traversal for a Graph, Depth First Traversal for a Graph, Eulerian path and circuit, Eulerian path for an undirected graph, Eulerian circuit for an undirected graph, Hamiltonian path in an undirected graph

Unit VI

Problems and algorithms based on Graphs : Dijkstra's shortest path algorithm, Bellman Ford algorithm, Minimum Spanning Tree algorithms namely Prim's algorithm, Kruskal algorithm, Graph colouring problem, Check if graph is bipartite, Detect cycle in graph, Check if cycle exists between two nodes, Strongly Connected Components, Shortest path from 1 to n, Minimum product spanning tree, Distance of nearest cell having 1, Count number of trees in a forest, Snake and Ladder Problem

List of Practicals / Experiments:

Practicals

- Program to implement Min Heap
- Program to implement Max Heap
- Program to print K'th largest element in array
- Program to connect n ropes with minimum cost
- Program to sort an almost sorted array
- Program to insert an element in binary tree
- Program to delete an element from a tree
- Program to implement tree traversals such as inorder, preorder and postorder

- Program to populate inorder successor of all nodes
- Program to implement boundary traversal of all nodes
- Program to find lowest common ancestor
- Program to print level order traversal in spiral form
- Program to implement BFS of a graph
- Program to implement DFS of a graph
- Program to implement Dijkstra's shortest path algorithm
- Program to implement Bellman Ford algorithm
- Program for graph colouring problem
- Program to check if graph is bipartite
- Program to detect cycle in graph

References:

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