BCSC1012: DESIGN & ANALYSIS OF ALGORITHMS

Objective: *The objective of this course is that students will construct and application of various data structures and concepts including Trees, Recursion & Dynamic programing.*

Credits:03 **L-T-P-J:3-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction:** Algorithms, analyzing algorithms, Complexity of algorithms, Growth of functions, Performance measurements, Sorting and order Statistics - Shell sort, Heap sort. Divide-Conquer approach with Quick sort, Merge sort. Comparison of sorting algorithms, Sorting in linear time. **Advanced Data Structures:** Red-Black Tree-Properties, Insertion, B Trees-Creation, Insertion, and Deletion. Introduction to Binomial Heaps-Merge, Union Operation, Fibonacci Heaps: Insertion, Finding Minimum Key, Union, Amortized Cost Graph Traversal-Breadth First Search, Depth First Search	20
II	Greedy methods-Fractional Knapsack, Activity Selection Problem. Minimum Spanning trees – Prim's and Kruskal's algorithms. Single source shortest paths - Dijkstra's and Bellman Ford algorithms. Dynamic programming-0/1Knapsack, LCS, Matrix Chain Multiplication. Backtracking, Branch and Bound with examples such as Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cycles and Sum of subsets. All pair shortest paths – Warshal's and Floyd's algorithms Introduction to P, NP, NP-complete, NP-Hard	20

Text Books:

- Thomas H. Coremen, Charles E. Leiserson and Ronald L. Rivest, Introduction to Algorithms, Third edition, Prentice Hall of India,2008.

Reference Books:

- Gilles Brassard Paul Bratley," Fundamentals of Algorithms", Prentice Hall,1996.
- Ellis Horowitz, SartajSahni, SanguthevarRajasekaran, "Fundamentals of Computer Algorithms", Orient Longman Pvt. Ltd,2008.
- Levitin , "An Introduction to Design and Analysis of Algorithms", Pearson,2008.

Outcome: After completion of course, student will be able to:

- CO1: Understanding of complexity representation in terms of Big Oh, Theta and Omega notations.
- CO2: Derive and solve recurrences describing the performance of divide-and-conquer algorithms (quick sort and merge sort).
- CO3: Compare and analyze different data structures (RB Tree, B Tree, Binomial Heaps, Fibonacci Heaps).
- CO4: Understand the major graph algorithms (DFS, BFS, Dijkstra's Bellman Ford) and their analyses.
- CO5: Understand the greedy paradigm and able to analyze when an algorithmic design situation calls for it. Synthesize greedy algorithms (Optimal Reliability Allocation, Minimum Spanning Trees, factorial Knapsack) and analyze them.
- CO6: Synthesize dynamic-programming algorithms (0/1 knapsack problem, Resource allocation problem, Warshal's and Floyd's algorithms) and analyze them.
- CO7: Understand the backtracking paradigm and able to analysis when an algorithmic design situation calls for it. Synthesize backtracking algorithms (N Queen Problem, TSP Problem, sum of subsets problem, Graph Coloring) and analyze them.
- CO8: Understand the branch and bound paradigm and able to analysis when an algorithmic design situation calls for it. Synthesize branch and bound algorithms (N Queen Problem, TSP Problem, Hamiltonian Cycles, Graph Coloring) and analyze them.

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs):

COs	POs/PSOs
CO1	PO1,PO3,PO4,PO12/PSO1,PSO3
CO2	PO1,PO3,PO4,PO5/PSO1,PSO3
CO3	PO1,PO3,PO6/PSO1,PSO3
CO4	PO1,PO2,PO3,/PSO1,PSO3
CO5	PO1,PO2/PSO1,PSO3
CO6	PO1,PO2,PO3, PO6/PSO1,PSO3
CO7	PO1,,PO4,PO12/PSO1,PSO3
CO8	PO1,PO2,PO3,PO4,PO12/PSO1,PSO2