BCA (HONS) 4th SEMESTER
DISCIPLINE SPECIFIC COURSE (CORE)

BCA420C1: DESIGN AND ANALYSIS OF ALGORITHMS

CREDITS: THEORY: 4; PRACTICAL: 2
MAX. MARKS: THEORY: 60; PRACTICAL: 30
MIN. MARKS: THEORY: 24; PRACTICAL: 12

THEORY: 60 LECTURES

UNIT-I

1. Introduction **(7 Lectures)**

Basic Design and Analysis techniques of Algorithms, Correctness of Algorithm.

2. Algorithm Design Techniques **(8 Lectures)**

Iterative techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.

UNIT-II

3. Sorting and Searching Techniques **(15 Lectures)**

Elementary sorting techniques-Bubble Sort, Insertion Sort, Merge Sort, Advanced Sorting techniques - Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Searching Techniques, Medians & Order Statistics, complexity analysis;

UNIT-III

4. Lower Bounding Techniques **(7 Lectures)**

Decision Trees

5. Balanced Trees **(8 Lectures)**

Red-Black Trees

UNIT-IV

6. Advanced Analysis Technique **(5 Lectures)**

Amortized analysis

7. Graphs **(5 Lectures)**

Graph Algorithms-Breadth First Search, Depth First Search and its Applications, Minimum Spanning Trees.

8. String Processing **(5Lectures)**

String Matching, KMP Technique

Recommended Books:

T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein Introduction to Algorithms, PHI, 3rd Edition 2009.
Sarabasse & A.V. Gelder Computer Algorithm -Introduction to Design and Analysis, Publisher - Pearson 3rd Edition 1999.

LAB: DESIGN AND ANALYSIS OF ALGORITHMS **(2 CREDITS; 60 LECTURES)**

1. i. implement Insertion Sort (The program should report the number of comparisons)
   ii. Implement Merge Sort (The program should report the number of comparisons)
2. Implement Heap Sort (The program should report the number of comparisons)
3. Implement Randomized Quick sort (The program should report the number of comparisons)
4. Implement Radix Sort
5. Create a Red-Black Tree and perform following operations on it:
   i. Insert a node
   ii. Delete a node
   iii. Search for a number & also report the color of the node containing this number.
6. Write a program to determine the LCS of two given sequences
7. Implement Breadth-First Search in a graph
8. Implement Depth-First Search in a graph
9. Write a program to determine the minimum spanning tree of a graph

For the algorithms at S. No 1 to 3 test run the algorithm on 100 different inputs of sizes varying from 30 to 1000. Count the number of comparisons and draw the graph. Compare it with a graph of nlogn.