Educate Elevate Enlighten

JSS Mahavidyapeetha
JSS Science And Technology University
(Established Under JSS Science and Technology University Act No. 43 of 2013)

JSS SCIENCE AND TECHNOLOGY UNIVERSITY MYSURU

Course Title: Design and Analysis of Algorithms Lab	***Course Code: 20CS47L***
Credits (L: T:P): 0:0:1.5	***Contact Hours (L: T: P): 0:0:39***
Type of Course: Practical	***Category: Professional Core Course***
CIE Marks: 50	***SEE Marks: 50***

Pre-requisite: Data Structures

Course Outcomes: After completing this course, students should be able to:

CO-1	Analyze the problem domain; Choose the appropriate data structures and design technique based on the problem domain.
CO-2	Implement algorithms and perform analysis with empirical method.
CO-3	Evaluate the performance of different algorithms using different design techniques for solving the same problem.

Unit No.	**List of Programs**
1	Implement Euclid’s, consecutive integer checking and modified Euclid’s algorithms to find GCD of two nonnegative integers and perform comparative analysis.
2	Implement the following searching algorithms and perform their analysis for worst case, best-case and average inputs. a) Sequential Search b) Binary Search(Recursive)
3	Implement the following elementary sorting algorithms and perform their analysis for worst case, best-case and average inputs(Note: Any two may be asked in the test/exam) a) Selection Sort b) Bubble Sort c) Insertion Sort
4	Implement Brute force string matching algorithm to search for a pattern of length ‘M’ in a text of length ‘N’ (M<=N) and perform its analysis for worst case, best-case and average inputs.
5	Implement Merge Sort algorithm and perform its analysis for worst case, best-case and average inputs.
6	Implement Quick Sort algorithm and perform its analysis for worst case, best-case and average inputs.
7	Implement DFS algorithm to check for connectivity and acyclicity of a graph. If not connected, display the connected components. Perform its analysis for different inputs

8	Implement BFS algorithm to check for connectivity and acyclicity of a graph. If not connected, display the connected components. Perform its analysis for different inputs
9	Implement DFS based algorithm to list the vertices of a directed graph in Topological ordering. Perform its analysis for different inputs
10	Implement source removal algorithm to list the vertices of a directed graph in Topological ordering. Perform its analysis for different inputs (Note: use efficient method to identify the source vertex).
11	Implement heap sort algorithm with bottom-up heap construction. Analyze its efficiency for worst case, best-case and average case inputs.
12	a) Implement Warshall's Algorithm to find the transitive closure of a directed graph and perform its analysis for different inputs. b) Implement Floyd's Algorithm to find All-pair shortest paths for a graph and perform its analysis for different inputs.
13	a) Implement an algorithm to solve Knapsack problem with bottom up Dynamic Programming approach and perform its analysis for different inputs. b) Implement a Dynamic Programming algorithm with Memory function to solve Knapsack problem with approach and perform its analysis for different inputs.
14	Implement Prim's algorithm to find Minimum Spanning Tree of a graph and perform its analysis for different inputs.