❖ ASSIGNMENT NUMBER 3:

		CO	BL
	Objective: • Understand and implement Divide and Conquer strategies. • Apply Greedy Method for optimization problems. • Analyse time complexity and behaviour of the algorithms through practical coding.		
Q.1	**Binary Search Implementation:** • **Task:** Write a program to perform binary search on a sorted array. • **Requirements:** • Implement both recursive and iterative versions. • Display the number of comparisons made. • **Input:** Sorted array of integers and a target element. • **Output:** Index of the element (or appropriate message if not found).	CO3	BL3
Q.2	**Quick Sort Implementation** • **Task:** Implement the Quick Sort algorithm. • **Requirements:** • Use a pivot selection strategy (first element, last element, or median). • Track the number of comparisons and swaps. • **Input:** Unsorted array of integers. • **Output:** Sorted array.	CO3	BL3
Q.3	**Strassen's Matrix Multiplication** • **Task:** Implement Strassen's algorithm for matrix multiplication. • **Requirements:** • Handle matrices of size 2^n × 2^n (padding if necessary). • Compare execution time with conventional matrix multiplication. • **Input:** Two matrices. • **Output:** Product matrix.	CO3	BL3
Q.4	**Greedy Method Algorithms** Minimum Cost Spanning Tree (MST) • **Task:** Implement Prim's and Kruskal's algorithms to find MST. • **Requirements:** • Print the edges included in the MST and total cost. • **Input:** A connected, weighted undirected graph (using adjacency matrix or adjacency list). **Output:** MST and total minimum cost.	CO3	BL3

By Assistant Professor: Megha Rathore

Q.5	**Knapsack Problem (Fractional)** • **Task:** Solve the Fractional Knapsack Problem using a greedy approach. • **Requirements:** • Items should be sorted based on value/weight ratio. • Allow taking fractions of an item. • Input: Arrays of weights, values, and the capacity of the knapsack. • Output: Maximum value that can be put in the knapsack.	CO3	BL3