S.No.	Implementation Details
LAB 1	**1.1 Sorting Algorithms** • Insertion Sort • Selection Sort • Quick Sort • Merge Sort • Bubble Sort • Randomised Quick Sort
	1.2 Searching Algorithms • Linear Search • Binary Search
	1.3 Problems ◦ Finding the Kth smallest element out of two sorted list of sizes M and N. ◦ Finding missing integers from a list of n-1 integers given the range of integers from 1 to n without any duplicates. ◦ Computing the count of 0s from the sorted array of large size (say M) containing elements as 0 and 1 only.
LAB 2	**2.1 Hybridised Quick Sort** ◦ Implementation of Hybridised Quick Sort to reduce the time complexity for sorting large size of array in which Insertion Sort Algorithm is applied upto size 10 and for size > 10, Quick Sort Algorithm is applied.
LAB 3	**3.1 Divide and Conquer Method** ◦ Implementation of Strassen's Multiplication Method & Naive Matrix Multiplication & compare them using square matrix for n = 3, 4, 5, 6, 7, 8 ◦ Implementation of multiplication of two n - bit numbers & Naive Multiplication Method and compare them in terms of time using n = 4, 8, 16, 32 and 64 ◦ Implementation of Maximum Value Contiguous Subsequence. ◦ Implementation of Algorithm (named as algo_1) which is basically the implementation of Stooge Sort
LAB 4	**4.1 Dynamic Programming** ◦ Implementation of LCS Algorithm for A[1…n] and B[1…n] sequences ◦ Implementation of Longest Increasing Subsequence for an array A[1…n] ◦ Implementation of Longest Alternating Subsequence for an array A[1…n] ◦ Implementation of Longest Palindrome Subsequence for an array A[1…n]
LAB 5	**5.1 Dynamic Programming Method** ◦ Implementation of Matrix Chain Multiplication for the given n matrix <M1*M2*…*Mn> where the size of the matrix is M1 = di-1*di. ◦ Implementation of Optimal Binary Search Tree for the given n keys (K1, K2, … Km) whose pi and qi (dummy keys) are given ◦ Implement 0/1 Knapsack Problem using Dynamic Programming.
LAB 6	**6.1 Graph Algorithm** ◦ Implement Breadth First Search (BFS) algorithm for a given Graph G ◦ Implement Depth First Search (DFS) algorithm for a given Graph G ◦ Implement Topological Sorting for a given Graph G ◦ Implement to find the strongly connected components in a Graph
LAB 7	**7.1 Graph Algorithm** ◦ Implement Prims Algorithm for a given Graph G ◦ Implement Kruskal's Algorithm for a given Graph G ◦ Implement Dijkstra Algorithm to find single source shortest path in a given Graph G