TOPIC WISE LISTING

S.No.	Implementation	Application

1.	SORTING
	    a) Bubble Sort
	    b) Insertion Sort
	    c) Selection Sort
	    d) Shell Sort
	    e) Quick Sort
	    f) Radix Sort
		a) Disk File Sort

2.	SEARCHING
	    a) Sequential Search
	    b) Binary Search
		a) Disk File Searching

3.	STACKS
	    a) Array implementation
	    b) Linked List implementation
		a) Infix to postfix (**not to be included in exam**)
		b) Evaluation of postfix expression
		c) DFS
		d) Balancing of parenthesis

4.	QUEUES
	Linked List implementation of
	    a) Ordinary Queue
	    b) Circular Queue
	    c) Double ended Queue
		a) BFS
		b) Priority Queues as a Linked List

5.	LINKED LISTS
	
	To be implemented on ONE list
	
	a) Singly Linked Lists
	  ▪ Insert, Display, Delete, Search, Count, Reverse, Sort
	b) Circular Linked List
	  ▪ Insert, Display, Delete, Search, Count, Reverse, Sort
	c) Doubly Linked Lists
	  ▪ Insert, Display, Delete, Search, Count, Reverse, Sort
	
	To be implemented on TWO lists
	
	d) Singly Linked Lists
	  ▪ Merge, Append, Union, Intersection
	e) Doubly Linked Lists
	  ▪ Merge, Append, Union, Intersection	

6.	BINARY SEARCH TREES
	
	Operations to be implemented on a BST
	  ▪ Insert
	  ▪ Recursive Traversals – postorder, inorder, preorder
	  ▪ Delete
	  ▪ Search
	  ▪ Largest Node
	  ▪ Smallest Node
	  ▪ Count number of nodes	

7.	HEAP
	
	    a) MinHeap
	
	        ▪ reheapUp, reheapDown, Delete
	
	    b) MaxHeap
	
	        ▪ reheapUp, reheapDown, Delete
		a) Heap Sort
		b) Select K (Example: Select 5 - to find the 5th largest element in a heap, perform 4 deletions on the heap and the element at the root will then be the 5th largest element)

8.	HASHING
	Methods for Hashing
	
	    a) Direct
	
	    b) Subtraction
	
	    c) Modulo Division
	
	    d) Digit Extraction
	
	    e) Fold Shift
	
	    f) Fold Boundary
	
	Method for Collision Resolution
	
	    a) Linear Probe	

9.	GRAPHS
	
	    a) Represent a graph using the Adjacency Matrix
	
	    b) Find the Shortest Path in a Graph using Warshall's Algorithm
	
	    c) Find the Minimum Spanning Tree (using ANY method Kruskal's Algorithm or Prim's Algorithm)	