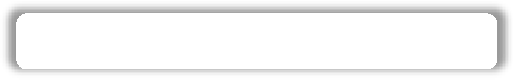
## JSPM’s Narhe Technical Campus

**MCA Department**

## MCA-I(Sem-I)

**(Data Structure and Algorithm)**



**DSA (Practical List)**

# Write a JS program for Singly Linked List-

* + To create Linked List.
  + Add the Node a)at beginning b)at end c)at specific location

# To display Linked List.

* + To search given data in Linked List.

# Write a JS program for Doubly Linked List-

* + To insert Node at last position of linked list.
  + Add the Node a)at beginning b)at end c)at specific location

# To delete node from specific position of linked list.

# Write a JS program for Doubly linked list-

* + Sort the linked list in ascending order.

# And display it.

# Write a JS program to create a singly linked list and count total number of nodes in it and display the result.

# Write a JS program for stack with array implementation-

* + To check is empty.

# To Peek.

* + To PUSH.

# and POP the stack.

# Write a JS program for array implementation of circular Queue for integers-

* + Insert.

# Delete.

* + Display.

# Write a JS program to reverse a string using stack.

# Write a JS program to check for balanced parentheses by using stacks.

# Write a JS program Implement Stack using Linked List-

* + push().

# pop().

* + peek().

# display().

# Write a JS program Implement Queue using Linked List-

# Enqueue

# Dqueue

# Display

# Write a JS program to Enqueue, Dequeue, Front and Rear the linear Queue using array.

# Write a program for implementing Hashing using Hash Table and Hash Function.

# Practical based on Brute Force technique.

# Write a JS program to sort an array elements using Merge sort method.

# Write a JS program to sort an array elements using Quick sort method.

# Write a JS program to search an element in array using Leaner search method.

# Write a JS program to search an element in array using Binary search method.

# Write a JS program for Binary Search Tree-

# - Create BST.

# - Insert element in BST.

# - Display.

# Write a JS program to read the adjacency matrix of direct graph and convert it into adjacency list.

# Write a JS program to print DFS and BFS traversal from a given graph.

# Write JS program to count indegree and outdegree of each node in graph.

# Practical based on Brute Force technique.

# Practical based on Greedy Algorithm-Prim’s/Kruskal’s algorithm.

# Implementation of Dynamic Programming- LCS, Regular Expression Matching

# Practical based on backtracking- N Queen’s problems