## LINKED LIST-

Introduction to Linked Lists

Linked List Implementation | Part 1

Linked List Implementation | Part 2

Static Linked List in C

Clone given Linked List

Delete Linked List

Pop operation in linked list

Insert given node into the correct sorted position in the given sorted linked list

Given a linked list, change it to be in sorted order

Split the nodes of the given linked list into front and back halves

Remove duplicates from a sorted linked list

Move front node of the given list to the front of the another list

Move even nodes to the end of the list in reverse order

Split given linked list into two lists where each list containing alternating elements from it

Construct a linked list by merging alternate nodes of two given lists

Merge given sorted linked lists into one

Merge Sort Algorithm for Singly Linked List

Intersection of two given sorted linked lists

Reverse linked list | Part 1 (Iterative Solution)

Reverse linked list | Part 2 (Recursive Solution)

Reverse every group of k nodes in given linked list

Find K’th node from the end in a linked list

Merge alternate nodes of two linked lists into the first list

Merge two sorted linked lists from their end

Delete every N nodes in a linked list after skipping M nodes

Rearrange linked list in specific manner in linear time

**By-Anushka Gupta**

Check if linked list is palindrome or not

Move last node to front in a given Linked List

Rearrange the linked list in specific manner

Detect Cycle in a linked list (Floyd’s Cycle Detection Algorithm)

Sort linked list containing 0’s, 1’s and 2’s

Stack Implementation using Linked List

Queue Implementation using Linked List

Remove duplicates from a linked list

Rearrange the linked list so that it has alternating high, low values

Rearrange a Linked List by Separating Odd Nodes from the Even Ones

Calculate height of a binary tree with leaf nodes forming a circular doubly linked list

**By-Anushka Gupta**