Linked List Data Structure - GeeksforGeeks

Source: https://www.geeksforgeeks.org/data-structures/linked-list/

Courses Tutorials Practice Jobs DSA Tutorial Interview Questions Quizzes Must Do Advanced DSA System Design Aptitude Puzzles Interview Corner DSA Python Technical Scripter 2026 Explore DSA Fundamentals Logic Building Problems Analysis of Algorithms Data Structures Array Data Structure String in Data Structure Hashing in Data Structure Linked List Data Structure Stack Data Structure Queue Data Structure Tree Data Structure Graph Data Structure Trie Data Structure Algorithms Searching Algorithms Sorting Algorithms Introduction to Recursion Greedy Algorithms Tutorial Graph Algorithms Dynamic Programming or DP Bitwise Algorithms Advanced Segment Tree Binary Indexed Tree or Fenwick Tree Square Root (Sqrt) Decomposition Algorithm Binary Lifting Geometry Interview Preparation Interview Corner GfG160 Practice Problem GeeksforGeeks Practice - Leading Online Coding Platform Problem of The Day - Develop the Habit of Coding DSA Course 90% Refund Linked List Data Structure Last Updated : 26 Jan, 2026 A linked list is a fundamental data structure in computer science. It mainly allows efficient insertion and deletion operations compared to arrays . Like arrays, it is also used to implement other data structures like stack, queue and deque. A linked list is a type of linear data structure individual items are not necessarily at contiguous locations. The individual items are called nodes and connected with each other using links. A node contains two things first is data and second is a link that connects it with another node. The first node is called the head node and we can traverse the whole list using this head and next links. Here’s the comparison of Linked List and Arrays Linked List: Data Structure: Non-contiguous Memory Allocation: Typically allocated one by one to individual elements Insertion/Deletion: Efficient Access: Sequential Array: Data Structure: Contiguous Memory Allocation: Typically allocated to the whole array Insertion/Deletion: Inefficient Access: Random Basics Singly Linked List Doubly Linked List Circular Linked List Applications and Advantages Operations Length of Linked List Print Linked List Search in a Linked List Linked List Insertion Deleting a given key Deleting at given position Delete a Linked List Nth Node from Start Nth Node from End Size of Doubly Linked List Easy Problems Remove every k-th node Middle of a Linked List Count Occurrences in a Linked List Circular Linked List Traversal Check if Circular Count Nodes in Circular List Deletion from a Circular Linked List Singly to circular Conversion Exchange first and last nodes in Circular Delete in a Doubly Linked List Reverse a Singly Linked List Reverse a Doubly Linked List Medium Problems Swap Nodes in Pairs Detect loop in a linked list Length of loop in linked list Design Browser History Remove duplicates from a sorted linked list Remove Duplicates from an Unsorted Linked List Intersection of two Sorted Linked Lists Partition a List QuickSort on Singly Linked List Split a Circular Linked List into two halves Merge Two Sorted Linked Lists Union and Intersection Merge Sort for Doubly Linked List Pairs with Sum in doubly linked list Insert in sorted way in doubly linked list Remove duplicates from an unsorted DLL Rotate a Linked List Rotate Doubly linked list by N nodes Delete a node with only its pointer given Segregate even and odd nodes Hard Problems Merge K Sorted Lists Intersection point of two Linked Lists. Implement LRU Cache Clone a linked list with random pointer Binary Tree to Doubly Linked List Reverse a Singly Linked List in Groups Reverse a Doubly Linked List in Groups Sublist Search Linked list from 2D matrix Rotate Linked List block wise Multiply two numbers as Lists Delete N nodes after M nodes Quick Links : Practice Linked List Quiz on Linked List DSA Tutorial Comment Article Tags: Article Tags: Linked List DSA Data Structures