

[Data Structures](#)

[Algorithms](#)

[Linked List Data Structure](#)

[Interview Preparation](#)

[Last Updated : 22 Apr, 2022](#)

[Topic-wise Practice](#)

[Data Structure and Algorithms Course](#)

[C++ Practice Problems on Linked List](#)

[Recent Articles on Linked List](#)

[Java](#)

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The

[Python](#)

elements in a linked list are linked using pointers as shown in the below image:

[Competitive Programming](#)

[Machine Learning](#)

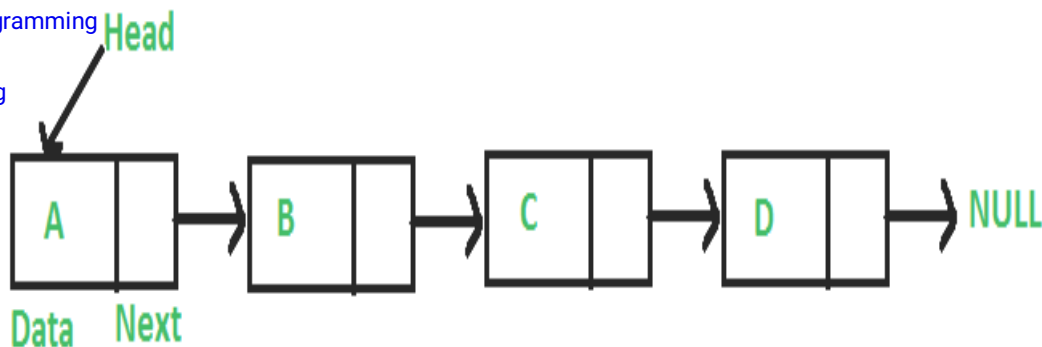
[HTML](#)

[SDE Sheet](#)

[Puzzles](#)

[GFG School](#)

[Projects](#)



In simple words, a linked list consists of nodes where each node contains a data field and a reference(link) to the next node in the list.

Topics :

- [Singly Linked List](#)
- [Circular Linked List](#)
- [Doubly Linked List](#)
- [Misc](#)
- [Quick Links](#)

Singly Linked List :

1. [Introduction to Linked List](#)
2. [Linked List vs Array](#)
3. [Linked List Insertion](#)
4. [Linked List Deletion \(Deleting a given key\)](#)
5. [Linked List Deletion \(Deleting a key at given position\)](#)
6. [Write a function to delete a Linked List](#)
7. [Find Length of a Linked List \(Iterative and Recursive\)](#)
8. [Search an element in a Linked List \(Iterative and Recursive\)](#)
9. [Write a function to get Nth node in a Linked List](#)
10. [Nth node from the end of a Linked List](#)
11. [Print the middle of a given linked list](#)
12. [Write a function that counts the number of times a given int occurs in a Linked List](#)
13. [Detect loop in a linked list](#)
14. [Find length of loop in linked list](#)
15. [Function to check if a singly linked list is palindrome](#)
16. [Remove duplicates from a sorted linked list](#)

17. [Remove duplicates from an unsorted linked list](#)
18. [Swap nodes in a linked list without swapping data](#)
19. [Pairwise swap elements of a given linked list](#)
20. [Move last element to front of a given Linked List](#)
21. [Intersection of two Sorted Linked Lists](#)
22. [Intersection point of two Linked Lists.](#)
23. [QuickSort on Singly Linked List](#)
24. [Segregate even and odd nodes in a Linked List](#)
25. [Reverse a linked list](#)

[More >>](#)

Circular Linked List :

1. [Circular Linked List Introduction and Applications,](#)
2. [Circular Linked List Traversal](#)
3. [Split a Circular Linked List into two halves](#)
4. [Sorted insert for circular linked list](#)
5. [Check if a linked list is Circular Linked List](#)
6. [Convert a Binary Tree to a Circular Doubly Link List](#)
7. [Circular Singly Linked List | Insertion](#)
8. [Deletion from a Circular Linked List](#)
9. [Circular Queue | Set 2 \(Circular Linked List Implementation\)](#)
10. [Count nodes in Circular linked list](#)
11. [Josephus Circle using circular linked list](#)
12. [Convert singly linked list into circular linked list](#)
13. [Circular Linked List | Set 1 \(Introduction and Applications\)](#)
14. [Circular Linked List | Set 2 \(Traversal\)](#)
15. [Implementation of Deque using circular array](#)
16. [Exchange first and last nodes in Circular Linked List](#)

[More >>](#)

Doubly Linked List :

1. [Doubly Linked List Introduction and Insertion](#)
2. [Delete a node in a Doubly Linked List](#)
3. [Reverse a Doubly Linked List](#)
4. [The Great Tree-List Recursion Problem.](#)
5. [Copy a linked list with next and arbit pointer](#)
6. [QuickSort on Doubly Linked List](#)
7. [Swap Kth node from beginning with Kth node from end in a Linked List](#)
8. [Merge Sort for Doubly Linked List](#)
9. [Create a Doubly Linked List from a Ternary Tree](#)
10. [Find pairs with given sum in doubly linked list](#)
11. [Insert value in sorted way in a sorted doubly linked list](#)
12. [Delete a Doubly Linked List node at a given position](#)
13. [Count triplets in a sorted doubly linked list whose sum is equal to a given value x](#)
14. [Remove duplicates from a sorted doubly linked list](#)
15. [Delete all occurrences of a given key in a doubly linked list](#)
16. [Remove duplicates from an unsorted doubly linked list](#)
17. [Sort the biotonic doubly linked list](#)
18. [Sort a k sorted doubly linked list](#)
19. [Convert a given Binary Tree to Doubly Linked List | Set](#)

20. [Program to find size of Doubly Linked List](#)
21. [Sorted insert in a doubly linked list with head and tail pointers](#)
22. [Large number arithmetic using doubly linked list](#)
23. [Rotate Doubly linked list by N nodes](#)
24. [Priority Queue using doubly linked list](#)
25. [Reverse a doubly linked list in groups of given size](#)
26. [Doubly Circular Linked List | Set 1 \(Introduction and Insertion\)](#)
27. [Doubly Circular Linked List | Set 2 \(Deletion\)](#)

[More >>](#)

Misc :

1. [Skip List | Set 1 \(Introduction\)](#)
2. [Skip List | Set 2 \(Insertion\)](#)
3. [Skip List | Set 3 \(Searching and Deletion\)](#)
4. [Reverse a stack without using extra space in O\(n\)](#)
5. [An interesting method to print reverse of a linked list](#)
6. [Linked List representation of Disjoint Set Data Structures](#)
7. [Sublist Search \(Search a linked list in another list\)](#)
8. [How to insert elements in C++ STL List ?](#)
9. [Unrolled Linked List | Set 1 \(Introduction\)](#)
10. [A Programmer's approach of looking at Array vs. Linked List](#)
11. [How to write C functions that modify head pointer of a Linked List?](#)
12. [Given a linked list which is sorted, how will you insert in sorted way](#)
13. [Can we reverse a linked list in less than O\(n\)?](#)
14. [Practice questions for Linked List and Recursion](#)
15. [Construct a Maximum Sum Linked List out of two Sorted Linked Lists having some Common nodes](#)
16. [Given only a pointer to a node to be deleted in a singly linked list, how do you delete it?](#)
17. [Why Quick Sort preferred for Arrays and Merge Sort for Linked Lists?](#)
18. [Squareroot\(n\)-th node in a Linked List](#)
19. [Find the fractional \(or \$n/k - th\$ \) node in linked list](#)
20. [Find modular node in a linked list](#)
21. [Construct a linked list from 2D matrix](#)
22. [Find smallest and largest elements in singly linked list](#)
23. [Arrange consonants and vowels nodes in a linked list](#)
24. [Partitioning a linked list around a given value and If we don't care about making the elements of the list "stable"](#)
25. [Modify contents of Linked List](#)

Quick Links :

- ['Practice Problems' on Linked List](#)
- ['Videos' on Linked List](#)
- ['Quizzes' on Linked List](#)

If you still need more assistance with your placement preparation, have a look at our [Complete Interview Preparation Course](#). The course has been designed by our expert mentors to help students **crack the coding interview of top product or service-based organizations** . You get access to **premium lectures, 200+ coding questions bank, resume building tips, and lifetime access** to the course content. So to make sure that your next programming interview doesn't feel like an interrogation, enroll in [Complete Interview Preparation](#) and give a boost to your placement preparation.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments



WHAT'S NEW



Must Do Coding Questions - Self Paced

[View Details](#)



Data Structures & Algorithms- Self Paced Course

[View Details](#)



Complete Interview Preparation

[View Details](#)

MOST POPULAR ARTICLES

[Best Time to Buy and Sell Stock](#)

[Must Do Coding Questions for Product Based Companies](#)

[Axios in React: A Guide for Beginners](#)

[How to Install Flutter on Visual Studio Code?](#)

[How to calculate MOVING AVERAGE in a Pandas DataFrame?](#)

MOST VISITED ARTICLES

[Retrofit with Kotlin Coroutine in Android](#)

[Difference between var, let and const keywords in JavaScript](#)

[Array of Objects in C++ with Examples](#)

[Router Configuration With Cisco Packet Tracer](#)

[Python Raise Keyword](#)



5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org



Company

[About Us](#)

[Careers](#)

[In Media](#)

[Contact Us](#)

[Privacy Policy](#)

[Copyright Policy](#)

Learn

[Algorithms](#)

[Data Structures](#)

[SDE Cheat Sheet](#)

[Machine learning](#)

[CS Subjects](#)

[Video Tutorials](#)

News

[Top News](#)

[Technology](#)

[Work & Career](#)

[Business](#)

[Finance](#)

[Lifestyle](#)

Languages

[Python](#)

[Java](#)

[CPP](#)

[Golang](#)

[C#](#)

[SQL](#)

Web Development

[Web Tutorials](#)

[Django Tutorial](#)

[HTML](#)

[CSS](#)

[JavaScript](#)

[Bootstrap](#)

Contribute

[Write an Article](#)

[Improve an Article](#)

[Pick Topics to Write](#)

[Write Interview Experience](#)

[Internships](#)

[Video Internship](#)

Start Your Coding Journey Now!

Login

Register