Introduction to Linked Lists

A linked list is a linear data structure consisting of a group of nodes where each node point to the next node through a pointer. Each node is composed of data and a reference (in other words, a link) to the next node in the sequence.

A linked list whose nodes contain two fields: an integer value and a link to the next node. The last node is linked to a terminator used to signify the end of the list.

Linked lists are among the simplest and most common data structures. The principal benefits of a linked list over a conventional array are:

- The list elements can easily be inserted or removed without reallocation or reorganization of the entire structure because the data items need not be stored contiguously in memory. Simultaneously, an array has to be declared in the source code before compiling and running the program.
- Linked lists allow the insertion and removal of nodes at any point in the list. They can do so with a constant number of operations if the link to the previous node is maintained during the list traversal.

On the other hand, simple linked lists by themselves do not allow random access to the data or any form of efficient indexing. Thus, many basic operations – such as obtaining the last node of the list, finding a node containing a given data, or locating the place where a new node should be inserted – may require sequential scanning of most or all of the list elements.

This website uses cookies. By using this site, you agree to the use of cookies, our policies, copyright terms and other conditions. Read our **Privacy Policy**. Got it

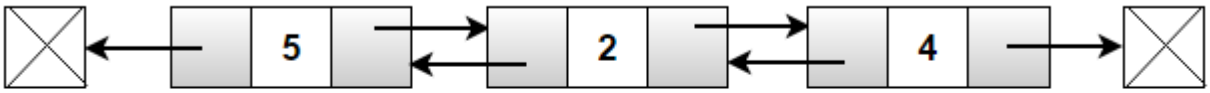
1. Singly Linked List

Singly linked lists contain nodes with a data field and a `next` field, which points to the next node in a line of nodes. The operations that can be performed on singly linked lists include insertion, deletion, and traversal.

*A **singly linked list** whose nodes contain two fields: an integer value and a link to the next node.*

2. Doubly Linked List

In a doubly-linked list, each node contains, besides the next-node link, a second link field pointing to the `prev` node in the sequence.

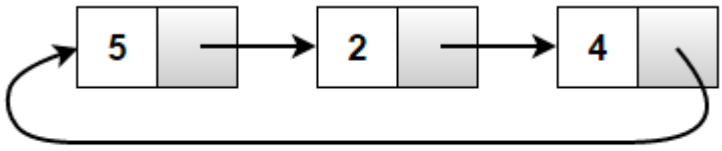


*A **doubly linked list** whose nodes contain three fields: an integer value, the link forward to the next node, and the link backward to the previous node*

An **XOR-linking technique** allows a doubly-linked list to be implemented using a single link field in each node.

3. Circular Linked list

In the last node of a list, the link field often contains a null reference, a special value is used to indicate the lack of further nodes. In a circular doubly linked list, the “tail” is linked back to the “head”. The major advantage of circularly linked lists is their ability to traverse the full list beginning at any given node.



*A **Circular linked list** whose tail node points to the head node*

This website uses cookies. By using this site, you agree to the use of cookies, our policies, copyright terms and other conditions. Read our **Privacy Policy**.

Applications of Linked Lists

Linked lists are a dynamic data structure, which can grow and be pruned, allocating, and deallocating memory while the program is running.

- Dynamic data structures such as stacks and queues can be implemented using a linked list and several other common abstract data types, including lists and associative arrays.
- Many modern operating systems use doubly linked lists to maintain references to active processes, threads, and other dynamic objects.
- A hash table may use linked lists to store the chains of items that hash to the same position in the hash table.
- A binary tree can be seen as a type of linked list where the elements are themselves linked lists of the same nature. The result is that each node may include a reference to the first node of one or two other linked lists, which, together with their contents, form the subtrees below that node.

Note: Unless explicitly mentioned, a singly linked list will be referred to by a list or linked list throughout our website.

Also See:

Linked List Implementation in C

Linked List – Insertion at Tail | C, Java, and Python Implementation

Source: https://en.wikipedia.org/wiki/Linked_list

Linked List

Beginner, Must Know

Techie Delight </>

Resources Online IDE Company

This website uses cookies. By using this site, you agree to the use of cookies, our policies, copyright terms and other conditions. Read our **Privacy Policy**.