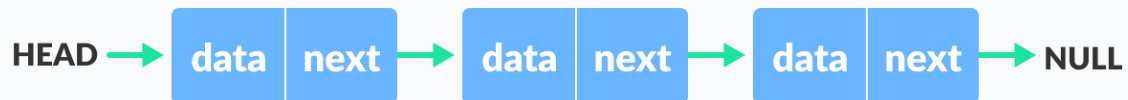


Linked list Data Structure

A linked list is a linear data structure that includes a series of connected nodes. Here, each node store the **data** and the **address** of the next node. For example,



Linked list Data Structure

You have to start somewhere, so we give the address of the first node a special name called `HEAD`. Also, the last node in the linked list can be identified because its next portion points to `NULL`.

Linked lists can be of multiple types: **singly**, **doubly**, and **circular linked list**.

Representation of Linked List

Each node consists:

- A data item
- An address of another node

Advantages of Linked List

1. The linked list is a dynamic data structure.
2. You can also decrease and increase the linked list at run-time. That is, you can allocate and deallocate memory at run-time itself.

3. In this, you can easily do insertion and deletion functions. That is, you can easily insert and delete the node
4. Memory is well utilized in the linked list. Because in it, we do not have to allocate memory in advance.
5. Its access time is very fast, and it can be accessed at a certain time without memory overhead.
6. You can easily implement linear data structures using the linked list like a stack, queue.

Disadvantages of Linked List

1. The linked list requires more memory to store the elements than an array, because each node of the linked list points a pointer, due to which it requires more memory.
2. It is very difficult to traverse the nodes in a linked list. In this, we cannot access randomly to any one node. (As we do in the array by index.) For example: – If we want to traverse a node in an n position, then we have to traverse all the nodes that come before n , which will spoil a lot of our time.
3. Reverse traversing in a linked list is very difficult, because it requires more memory for the pointer.

Application of Linked List

The linked list is a primitive data structure, which is used in various types of applications.

1. It is used to maintain directory names.
2. The linked list can perform arithmetic operations in the long integer.
3. Polynomials can be manipulated by storing constant in the node of the linked list.
4. We can also use it to next and previous images in the image viewer.
5. With the help of the linked list, we can move songs back and forth in the music player.
6. The linked list is also used for undo in word and Photoshop applications.
7. All the running applications in the computer are stored in the circular linked list, and the operating system provides them with a fixed time slot.
8. It can also be used to implement hash tables.

Difference between linked list and array

Both array and linked list are used to store the same type of linear data, but array is allocated contiguous memory location in the compile-time while the linked list is allocated memory in run-time.

Array	Linked List
It is a collection of the same type of data type.	It is a collection of similar elements that are connected to each other by the pointers.
It supports random access, which means that we can access it directly using its index, like arr[0] for 1st element, arr[7] for the 8th element, etc.	It supports sequential-access, meaning that in order to use any node in it, the user must traverse the whole list sequentially.
In the array, elements are stored in the contiguous-memory location.	The elements in the linked list can be stored anywhere in memory.
The time complexity of the array is $O(1)$.	The time complexity of the linked list is $O(n)$.
It is allocated the memory at compile-time.	It is allocated the memory at run-time.
Arrays take longer to perform insertion and deletion functions than linked lists.	In the linked list, both insertion and deletion operations take less time than the array.
It can be a 1-d array, 2-d array, or 3-d array.	It can be a linear linked list, doubly linked list, or circular linked list.