

Chapter 9

Learning objectives

- Linked list vs array comparison
- Application of linked list
- Advantages of linked list
- Disadvantages of linked list

Linked list vs array comparison

Having seen the operations and implementation of the linked list, let us compare how arrays and linked list fair in comparison with each other.\

Arrays	Linked lists
Arrays have fixed size	Linked list size is dynamic
Insertion of new element is expensive	Insertion/deletion is easier
Random access is allowed	Random access not possible
Elements are at contiguous location	Elements have non-contiguous location
No extra space is required for the next pointer	Extra memory space required for next pointer

Applications

As arrays and linked lists are both used to store items and are linear data structures, both these structures can be used in similar ways for most of the applications.

Some of the applications for linked lists are as follows:

- A linked list can be used to implement stacks and queues.
- A linked list can also be used to implement graphs whenever we have to represent graphs as adjacency lists.
- A mathematical polynomial can be stored as a linked list.
- In the case of hashing technique, the buckets used in hashing are implemented using the linked lists.
- Whenever a program requires dynamic allocation of memory, we can use a linked list as linked lists work more efficiently in this case.

Advantages Of Linked List:

Dynamic data structure: A linked list is a dynamic arrangement so it can grow and shrink at runtime by allocating and deallocating memory. So there is no need to give the initial size of the linked list.

No memory wastage: In the Linked list, efficient memory utilization can be achieved since the size of the linked list increase or decrease at run time so there is no memory wastage and there is no need to pre-allocate the memory.

Implementation: Linear data structures like stack and queues are often easily implemented using a linked list.

Insertion and Deletion Operations: Insertion and deletion operations are quite easier in the linked list. There is no need to shift elements after the insertion or deletion of an element only the address present in the next pointer needs to be updated.

Disadvantages Of Linked List:

Memory usage: More memory is required in the linked list as compared to an array. Because in a linked list, a pointer is also required to store the address of the next element and it requires extra memory for itself.

Traversal: In a Linked list traversal is more time-consuming as compared to an array. Direct access to an element is not possible in a linked list as in an array by index. For example, for accessing a node at position n , one has to traverse all the nodes before it.

Reverse Traversing: In a singly linked list reverse traversing is not possible, but in the case of a doubly-linked list, it can be possible as it contains a pointer to the previously connected nodes with each node. For performing this extra memory is required for the back pointer hence, there is a wastage of memory.

Random Access: Random access is not possible in a linked list due to its dynamic memory allocation.