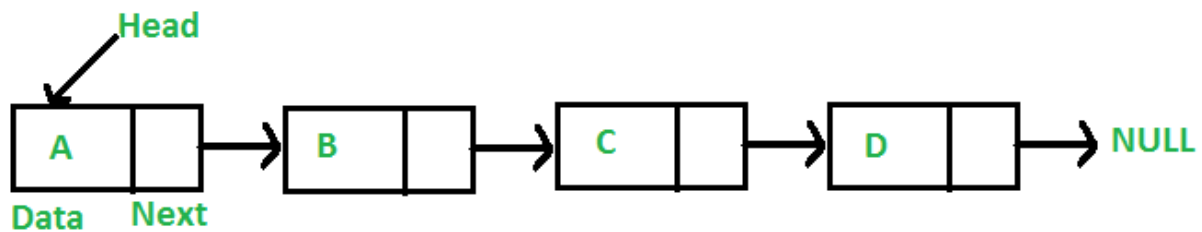


LINKED LIST

What is a linked list?

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers as shown in the below image:



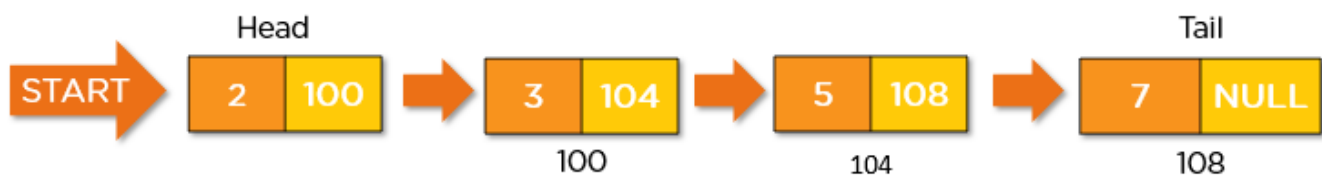
In simple words, a linked list consists of nodes where each node contains a data field and a reference(link) to the next node in the list.

What are the different forms of linked list?

There are four key types of linked lists:

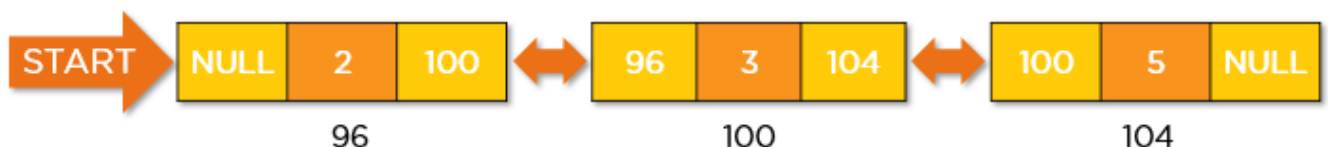
1) Singly linked lists

A singly linked list is a unidirectional linked list. So, you can only traverse it in one direction, i.e., from head node to tail node.



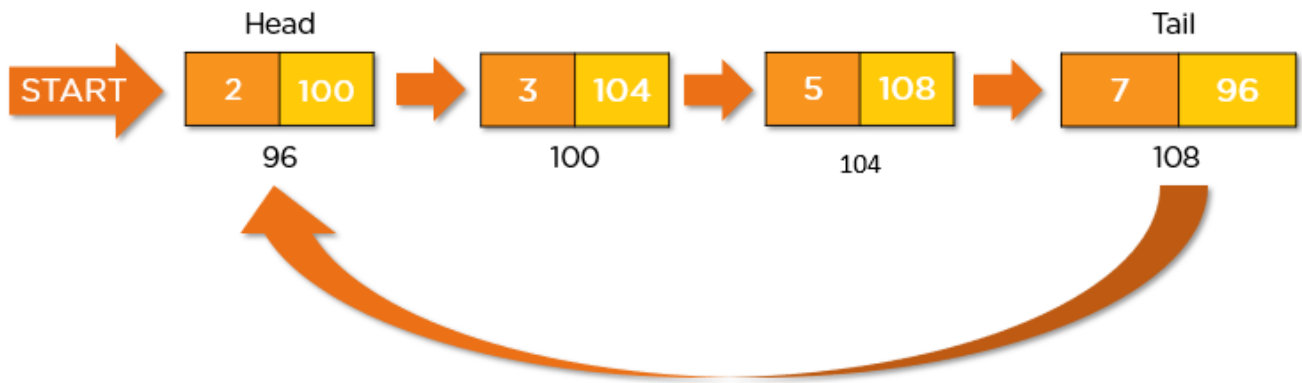
2) Doubly linked lists

A doubly linked list is a bi-directional linked list. So, you can traverse it in both directions. Unlike singly linked lists, its nodes contain one extra pointer called the previous pointer. This pointer points to the previous node.



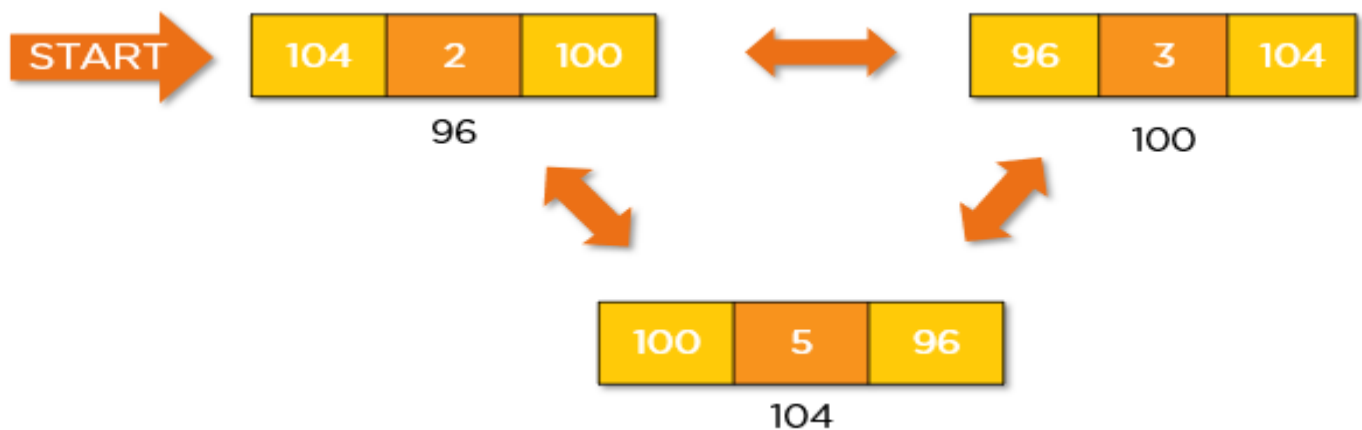
3) Circular linked lists

A circular linked list is a unidirectional linked list. So, you can traverse it in only one direction. But this type of linked list has its last node pointing to the head node. So while traversing, you need to be careful and stop traversing when you revisit the head node



4) Circular doubly linked lists

A circular doubly linked list is a mixture of a doubly linked list and a circular linked list. Like the doubly linked list, it has an extra pointer called the previous pointer, and similar to the circular linked list, its last node points at the head node. This type of linked list is the bi-directional list. So, you can traverse it in both directions.



What is a linked list's purpose?

- 1) Implementation of stacks and queues
- 2) Implementation of graphs: Adjacency list representations of graphs is most popular which uses linked list to store adjacent vertices.
- 3) Dynamic memory allocation: We use linked list of free blocks.
- 4) Maintaining directory of names
- 5) Performing arithmetic operations on long integers.

- 6) Manipulation of polynomials by storing constants in the node of linked list representing sparse matrices.

What are the advantages of linked lists over arrays?

- 1) Size is not an issue as compared to arrays.
- 2) Addition/Deletion of an element from the list at any index which is an $O(1)$ operation in Lists as compared to Arrays.
- 3) They can be used as underlying data structures for search trees, etc. I.e. you construct a search tree, whose lowest level consists of the linked list because lists express better properties as far as insertions are concerned.

What is the purpose of a circular linked list?

- 1) Entire list can be traversed from any node of the list.
- 2) It saves time when we have to go to the first node from the last node.
- 3) Its is used for the implementation of queue.
- 4) Reference to previous node can easily be found.
- 5) When we want a list to be accessed in a circle or loop then circular linked list are used.

How will you explain Circular Linked List?

A circular doubly linked list is a mixture of a doubly linked list and a circular linked list. Like the doubly linked list, it has an extra pointer called the previous pointer, and similar to the circular linked list, its last node points at the head node. This type of linked list is the bi-directional list. So, you can traverse it in both directions.

