

Session - 1

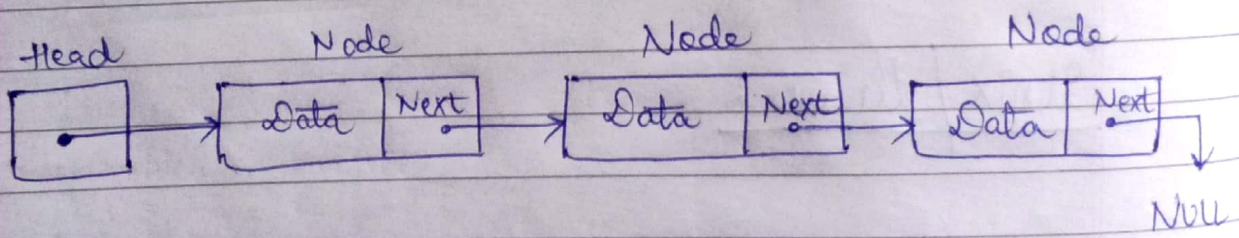
1. Linked Lists

- A linked list is a sequence of data structures, which are connected together via links.
- Linked list is a sequence of links which contains items

Link - Each link of a linked list can store a data called an element.

Next - Each link of a linked list contains a link to the next link called Next.

Linked List Representation -



- Linked list contains a link element called first
- Each link carries a data field and a link field called next
- Each link is linked with its next link using its next link.

Teacher's Signature.....

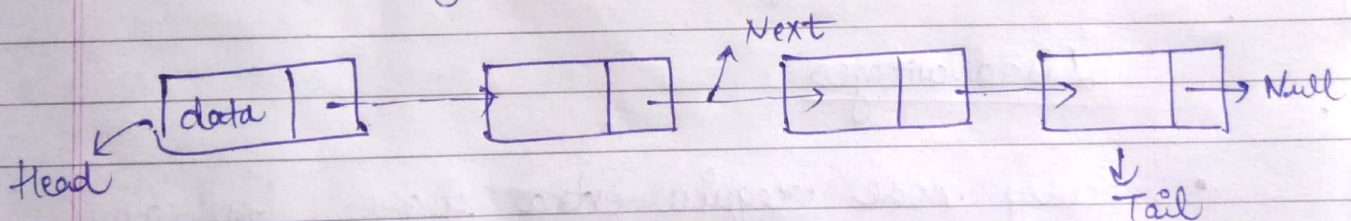
- Last link carries a link as null to mark the end of the list.

②. Singly Linked List

A singly linked list is a type of linked list that is unidirectional, that is it can be traversed only in one direction from head to the last node (tail).

Head → (First Node) It points to the first node of the list and help us access every other element in the list.

Tail → Points to null which helps us in determining when the list ends

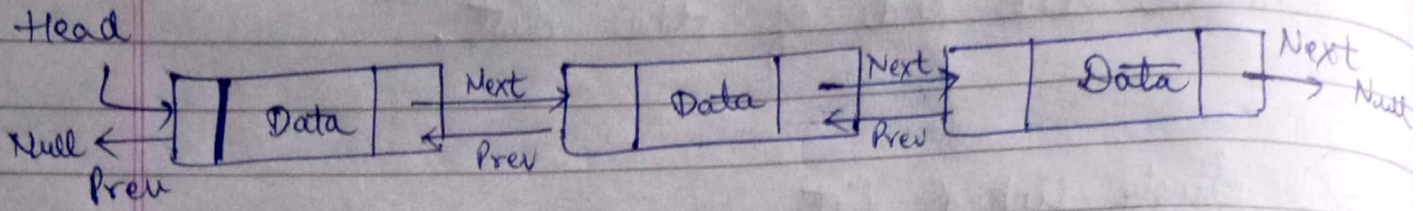


Example of singly linked list

③. Doubly Linked List

A doubly linked list contains an extra pointer, typically called previous pointer, together with next pointer and data which are there in singly linked list.

Teacher's Signature.....



Example of doubly linked list

Advantages

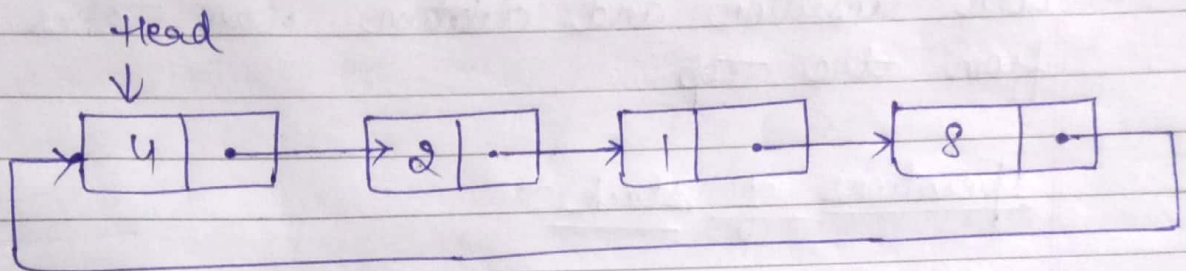
- Doubly linked list can be traversed in both dir. i.e forward and backward
- We can quickly insert a new node.
- The delete operation in DL is more efficient if pointer to the node to be deleted is given.

Disadvantages

- Every node require extra space for an previous pointer
- Extra operation is required. For example, in insertion, we need to modify previous pointer together with next pointer.

④ Circular Linked List

Circular linked list where all the nodes are connected to form a circle. There is no NULL at the end. A circular linked list can be singly circular linked list or doubly circular linked list



Example of circular linked list

Advantages -

- Any node can be starting point.
- Useful for implementation of queue.
- No requirement for a NULL assignment in the code.

Disadvantages -

- Circular list are complex.
- If not handled carefully, then the code go in a infinite loop.
- Harder to find end of the list.

Teacher's Signature.....

5. Stack

- Stack is a linear data structure
- It follows LIFO order (Last In first out)
- We can access only the top element that are available in the stack
- Both insertion and deletion can take place from the top

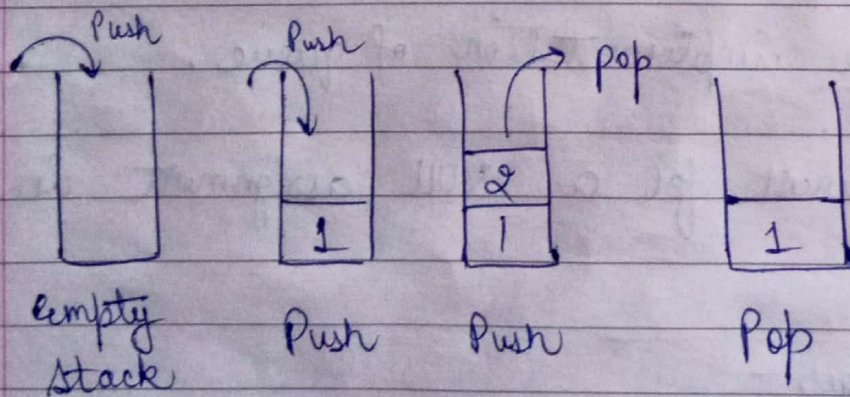
Operations of stack -

push() → To insert an element

pop() → To remove an element

isEmpty() → checks whether the stack is empty or not

peek() → To get the top most element



- ### Advantages -
- (i) Easy to start
 - (ii) Less Hardware Requirement
 - (iii) Cross-Platform

Disadvantages - (i) not flexible
(ii)
(iii)

Application - recursion, editors, browsers, parsing

(6) Queue -

- Linear data structure
- Follows first in first out (FIFO)
- Insertion takes place at rear end
- Deletion takes place from front end.

enqueue() → insert an element

dequeue() → To removes an element

peekfirst() → it will get the first element

peeklast() → To get the last element

Advantages - Maintains data in FIFO manner

Applications -

- Scheduling
- Maintaining the playlist
- Interrupt Handling



