DATA STRUCTURES AND ALGORITHMS

LINKEDLIST

Java LinkedList

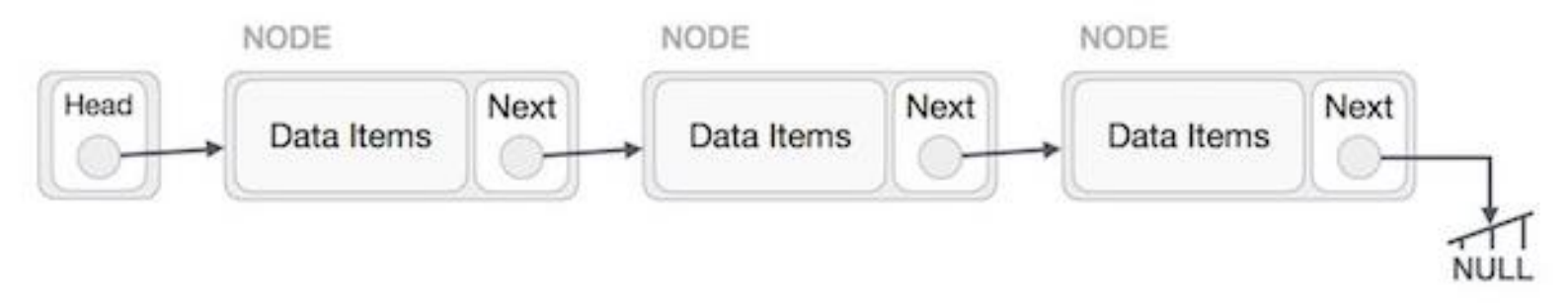
- A Linked List is a linear data structure which looks like a chain of nodes, where each node is a different element.
- A Linked List is, as the word implies, a list where the nodes are linked together. Each node contains data and a pointer. The way they are linked together is that each node points to where in the memory the next node is placed.

Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array. Following are the important terms to understand the concept of Linked List.

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.
LinkedList − A Linked List contains the connection link to the first link called First.

LINKED LIST REPRESENTATION

- Linked list can be visualized as a chain of nodes, where every node points to the next node.



- Linked list contains a link element called first.
- Each link carries a data field(s) and a link field called next.
- Each link is linked with its next link using its next link.
- Last link carries a link as null to mark the end of the list.

TYPES OF LINKED LISTS

1. SINGLY-LINKED LIST
2. DOUBLY LINKED LIST
3. CIRCULAR LINKED LIST

SINGLY-LINKED LIST

- Traversal of items can be done in the forward direction only due to the linking of every node to its next node.

```
public class Main {

    static class Node {
        int data;
        Node next;

        Node(int data) {
            this.data = data;
            this.next = null;
        }
    }

    public static void main(String[] args) {
        // Creating individual nodes
        Node firstNode = new Node(3);
        Node secondNode = new Node(5);
        Node thirdNode = new Node(13);
        Node fourthNode = new Node(2);

        // Linking nodes together
        firstNode.next = secondNode;
        secondNode.next = thirdNode;
        thirdNode.next = fourthNode;

        // Printing linked list
        Node currentNode = firstNode;
        while (currentNode != null) {
            System.out.print(currentNode.data + " -> ");
            currentNode = currentNode.next;
        }
        System.out.println("null");
    }
}
```

DOUBLY-LINKED LIST

- Traversal of items can be done in both forward and backward directions as every node contains an additional **prev** pointer that points to the previous node.

CIRCULAR-LINKED LIST

- A circular linked list is a type of linked list in which the first and the last nodes are also connected to each other to form a circle, there is no NULL at the end.

BASIC OPERATIONS

- **Insertion** − adds an element at the beginning of the list.
- **Deletion** − deletes an element at the beginning of the list.
- **Display** − displays the complete list.
- **Search** − searches an element using the given key.
- **Delete** − deletes an element using the given key.

LINKEDLIST METHODS

Method	Description
add(E e)	Adds the specified elements to the end of the list
Add(int index, E element)	Inserts the specified element at the specified position in the list
remove(int index)	Removes the element at the specified position in the list
remove(Object o)	Removes the first occurrence of the specified element from the list
get(int index)	Returns the element at the specified position in the list
set(int index, E element)	Replaces the element at the specified position with the specified element