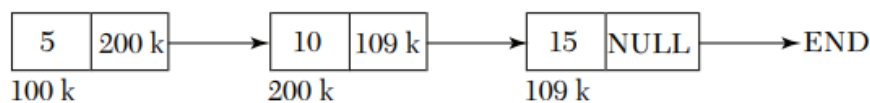


## SINGLY LINKED LIST(DS UNIT-II)

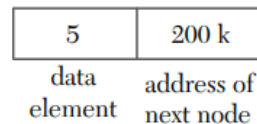
A linked list is a linear collection of data elements, called nodes, where the linear order is given by means of nodes. It is a dynamic data structure. For every data item in a linked list, there is an associated node that gives the memory location of the next data item in the linked list. The data items in the linked list are not in consecutive memory locations.

A linked list is a sequence of nodes in which each node contains one or more than one data field which points to the next node.



In the previous figure, we have made a linked list in which each node is divided into two parts:

1. The first part contains the information/data.
2. The second part contains the address of the next node.



### Advantages of linked lists

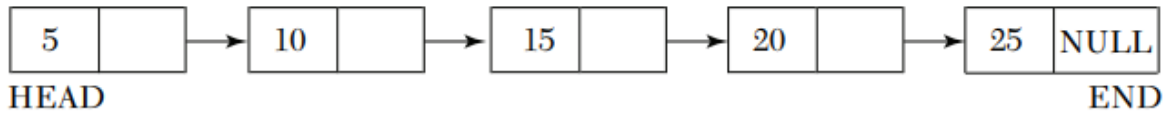
1. Linked lists are dynamic data structures; that is, they can grow or shrink during the execution of the program.
2. Linked lists have efficient memory utilization. Memory is allocated whenever it is required, and it is de-allocated whenever it is no longer needed.
3. Insertion and deletion are easier and efficient.
4. Many complex applications can be easily carried out with linked lists.

### Disadvantages of linked lists

1. They consume more space because every node requires an additional node to store the address of the next node.
2. Searching a particular element in the list is difficult and time-consuming

## Singly Linked List Implementation in JAVA

A singly linked list is the simplest type of linked list, in which each node contains some information/data and only one node which points to the next node in the linked list. The traversal of data elements in a singly linked list can be done only in one way.



### JAVA IMPLEMENTATION FOR INSERTION AND TRAVERSAL

```
class Node{
    int data;
    Node next;
}

class linklist{
    Node head;
    void insert(int d){
        Node nod=new Node();
        nod.data=d;
        nod.next=null;
        if(head==null)
            head=nod;
        else{
            Node n=head;
            while(n.next!=null){
                n=n.next;
            }
            n.next=nod;
        }
    }
    void show(){
        Node n=head;
        while(n.next!=null){
            System.out.print(n.data+"-->");
            n=n.next;
        }
        System.out.print(n.data);
    }
}
```

```

class driverlist{
    public static void main(String []args)
    {
        linklist l1=new linklist();
        l1.insert(5);
        l1.insert(10);
        l1.insert(15);
        l1.insert(20);
        l1.insert(25);
        l1.show();
    }
}

```

```

C:\javaprogram>java driverlist
5-->10-->15-->20-->25

```

#### NOTE

- A linked list is a sequence of nodes in which each node contains one or more than one data field and a node which points to the next node.
- The process of allocating memory during the execution of the program or the process of allocating memory to the variables at runtime is called dynamic memory allocation.
- A singly linked list is the simplest type of linked list, in which each node contains some information/data and only one node which points to the next node in the linked list.
- Traversing a linked list means accessing all the nodes of the linked list exactly once.
- Searching for a value in a linked list means to find a particular element/ value in the linked list.
- A circular linked list is also a type of singly linked list in which the address part of the last node will store the address of the first node. A doubly linked list is also called a two-way linked list; it is a special type of linked list which can point to the next node as well as the previous node in the sequence.
- A header linked list is a special type of linked list which always contains a special node, called the header node, at the beginning. This header node usually contains vital information about the linked list like the total number of nodes in the list, whether the list is sorted or not, and so forth.
- One of the most important applications of linked lists is polynomial representation, because linked lists can be used to represent polynomials, and there are different operations that can be performed on them.