

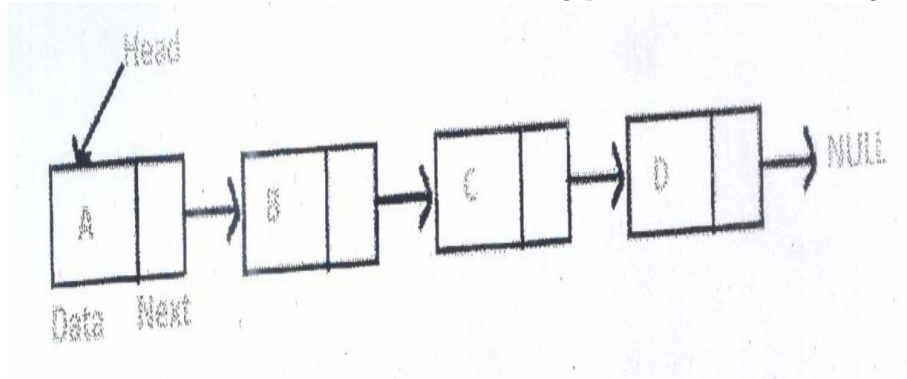
# DATA STRUCTURE = Linked List

soni200032

June 2019

## 1 Introduction

- \* A linked list is a linear data structure, in which the element are not stored at contiguous memory location.
  - \* It consists of group of nodes in a sequence.
- 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.
- \* Linked lists are used to create Trees and Graph.
  - \* The element in a linked list are linked using pointer As shown in Figure.



## 2 Advantages of Linked Lists

- \* They are dynamic in nature which allocates the memory when required.
- \* Insertion and deletion operation can be easily implemented.
- \* Linked List reduces the access time.

## 3 Disadvantages of Linked Lists

- \* The memory is wasted as pointer require extra memory for storage.
- \* No element can be accessed randomly; it has to access each node sequentially.
- \* Reverse Traversing is difficult in linked list.

## 4 Applications of Linked Lists

- \* Linked lists are used to implement stacks, queues, graphs, etc.
- \* Linked lists let you insert elements at the beginning and end of the list.
- \* In Linked Lists we don't need to know the size in advance.

## 5 The Operations on the Linked Lists

- \* Search: This operation involves the searching of an element in the linked list.
- \* Insertion: To add new node to data structure.
- \* Deletion: To delete a node from data structure.
- \* Merge: To merge two structures or more to constituting one structure.

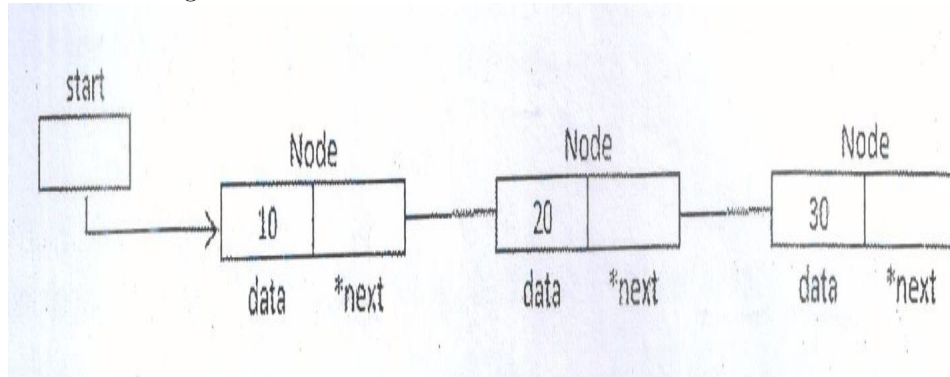
## 6 Types Of Linked Lists

There are four different implementations of linked list available, they are as follows.

- 6.1. Singly Linked List.
- 6.2. Doubly Linked List.
- 6.3. Circular Singly Linked List.
- 6.4. Circular Doubly Linked List.

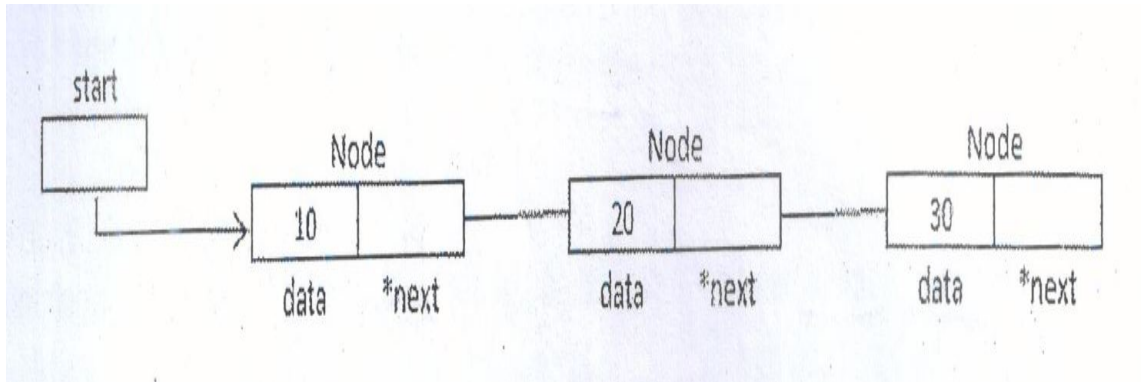
### 6.1 Singly Linked List

- \* Singly linked list contain nodes which have a data part as well as an address part i.e, next: which points to the next node in sequence of nodes.
- \* The operation we can perform on singly linked list are insertion, and deletion, As shown in Figure.



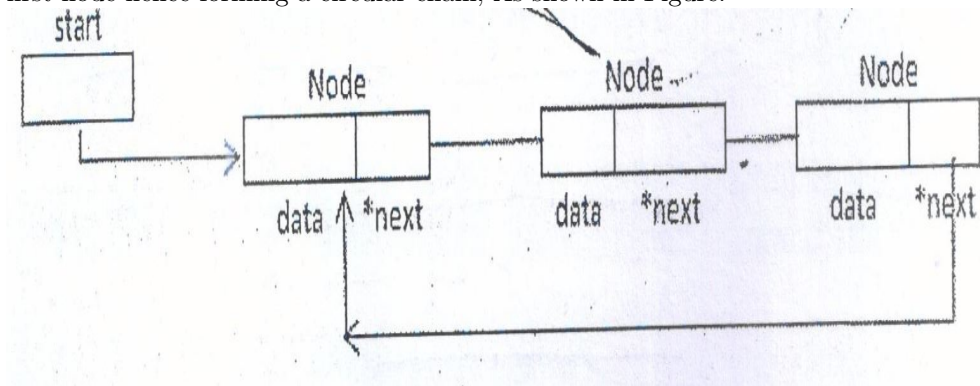
### 6.2 Doubly Linked List

- \* In a doubly linked list, each node contains a data part and two addresses, one for the previous node and other one for the next node, As shown in Figure.



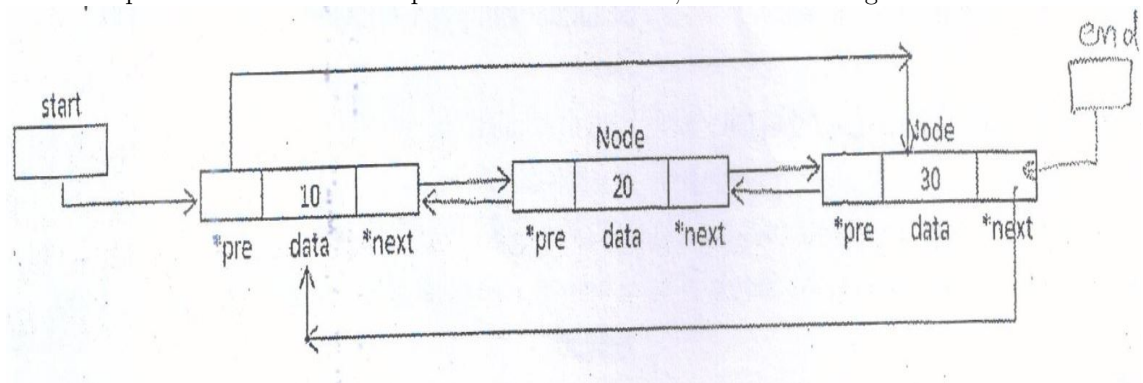
### 6.3 Circular Singly Linked List

\* In circular singly linked list the last node of the list holds the address of the first node hence forming a circular chain, As shown in Figure.



### 6.4 Circular Doubly Linked List

\* In circular doubly linked list the forward node points to the first node and backward pointer of the first node points to the last node, As shown in Figure.



A very helpfull reference is [1].

## References

- [1] Paul F Dietz. Maintaining order in a linked list. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 122–127. ACM, 1982.