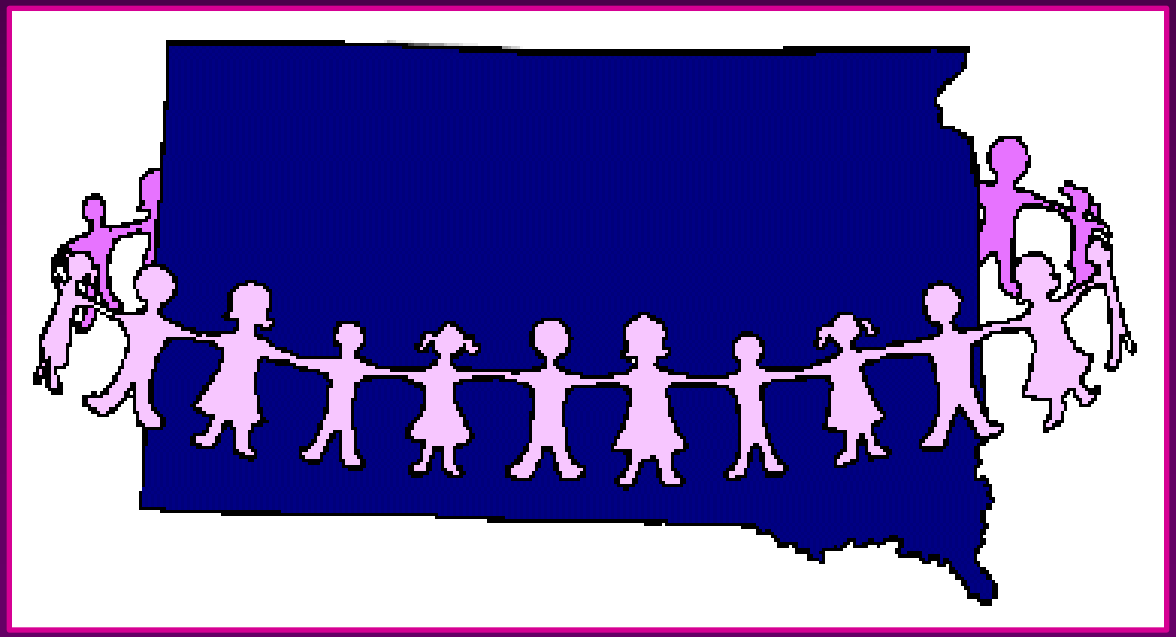
Circular Linked Lists - [CLL]



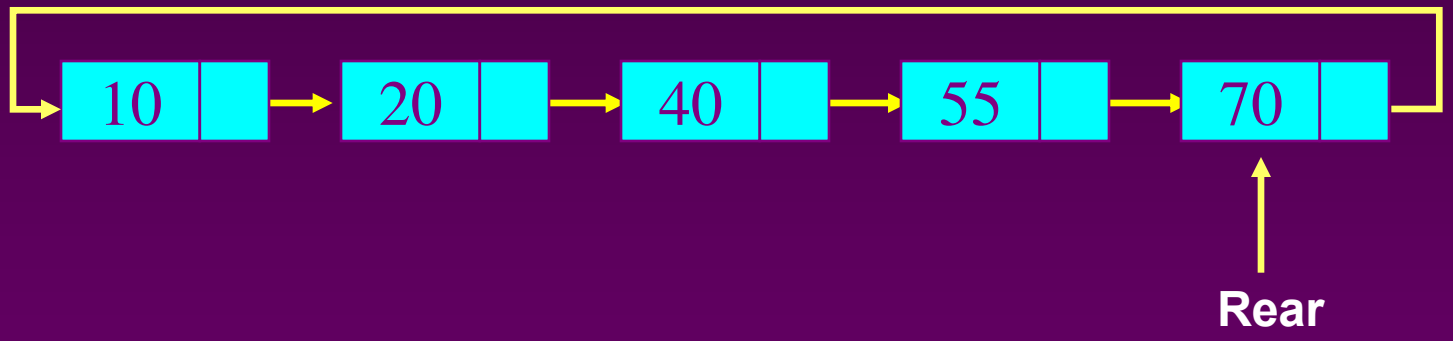
Types of Linked List

There are mainly three types of linked list

- **Singly linked list**
  - Each node has only one link part that contains the address of next node.
- **Circular linked list**
  - In this linked list the linked field of the last node contain the address of the first node of list
- **Doubly linked list**
  - In this linked list all nodes are linked together by multiple number of links which help in accessing both the successor and predecessor node from the given node position

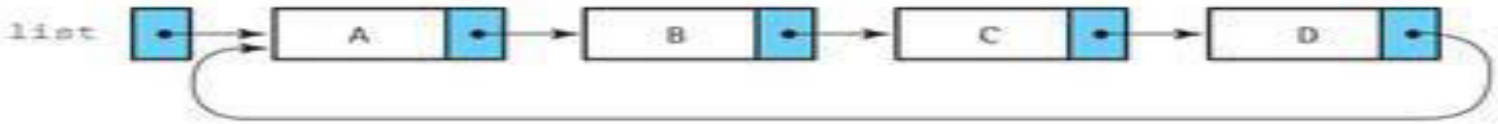
Circular Linked Lists

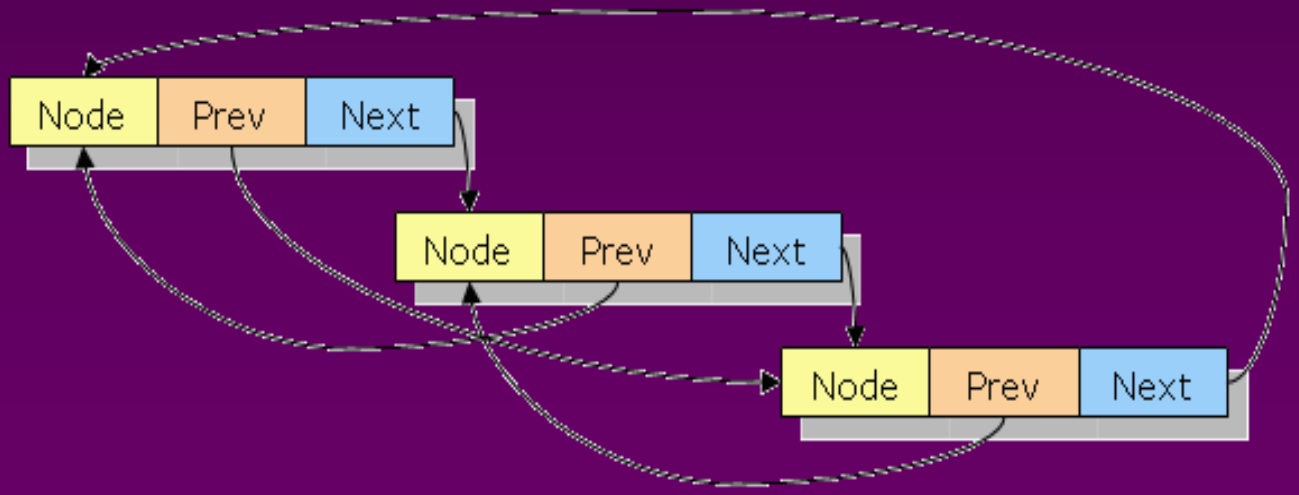
- A Circular Linked List is a special type of *Linked List.*
- It supports traversing from the *end of the list to the beginning by making the last node point back to the head of the list.*
- A Rear pointer is often used instead of a Head pointer



Circular Linked Lists

- **Circular linked list** A list in which every node has a successor; the "last" element is succeeded by the "first" element



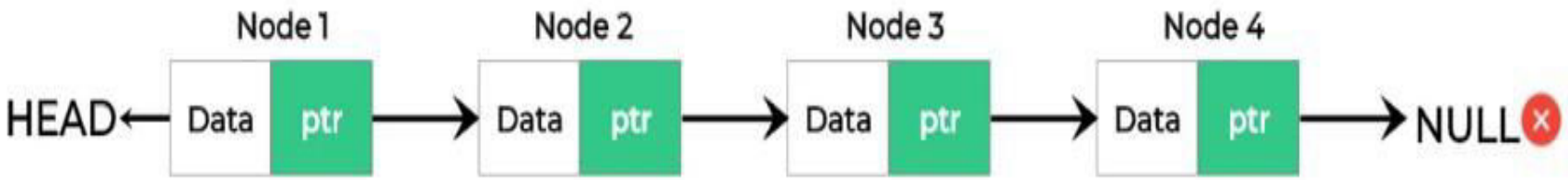


Motivation

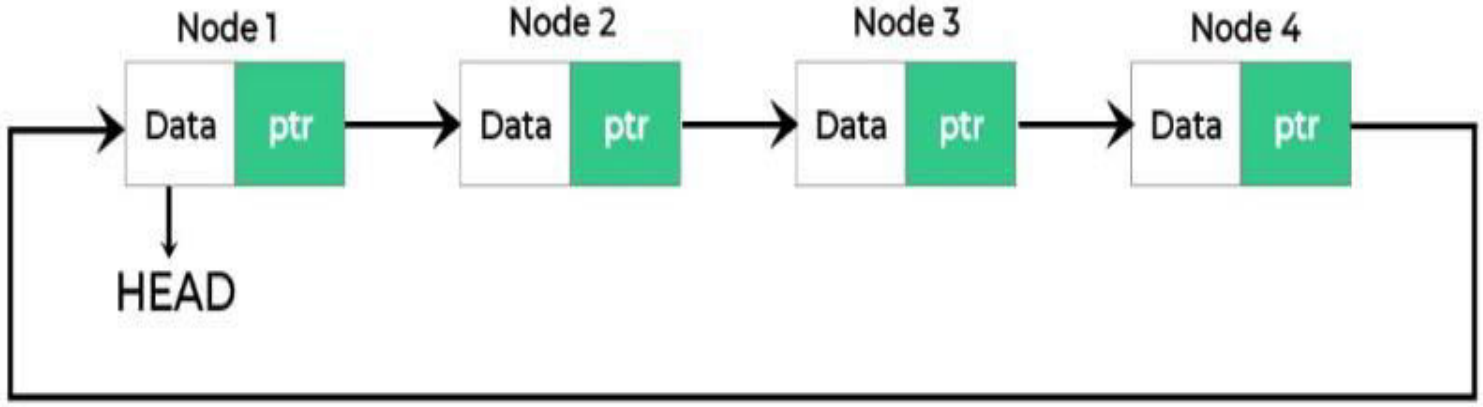
- Circular linked lists are *usually sorted.*
- Circular linked lists are useful for playing video and sound files in "looping" mode.
- **circularly linked list**, all nodes are **linked** in a continuous circle, *without using null.*
- For **lists** with a *front and a back (such as a queue), one stores a reference to the last node in the **list**.*
- The *next node after* the *last node is the first node.*

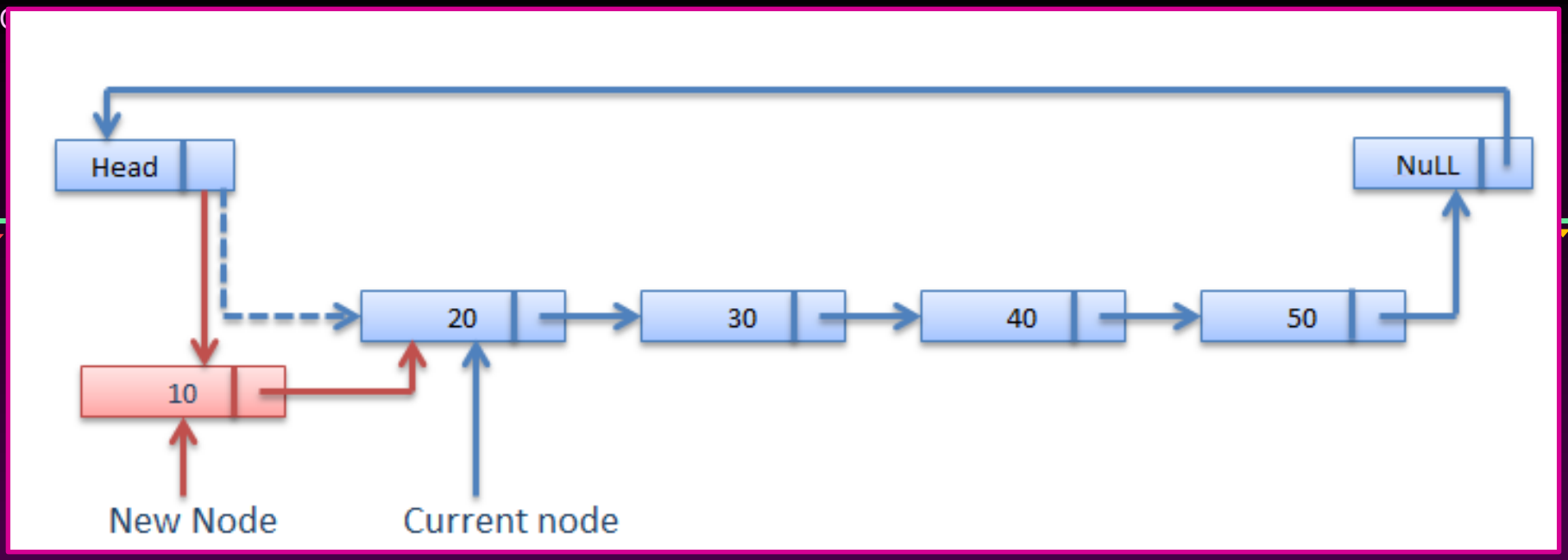
Difference between Linked List and Circular Linked List

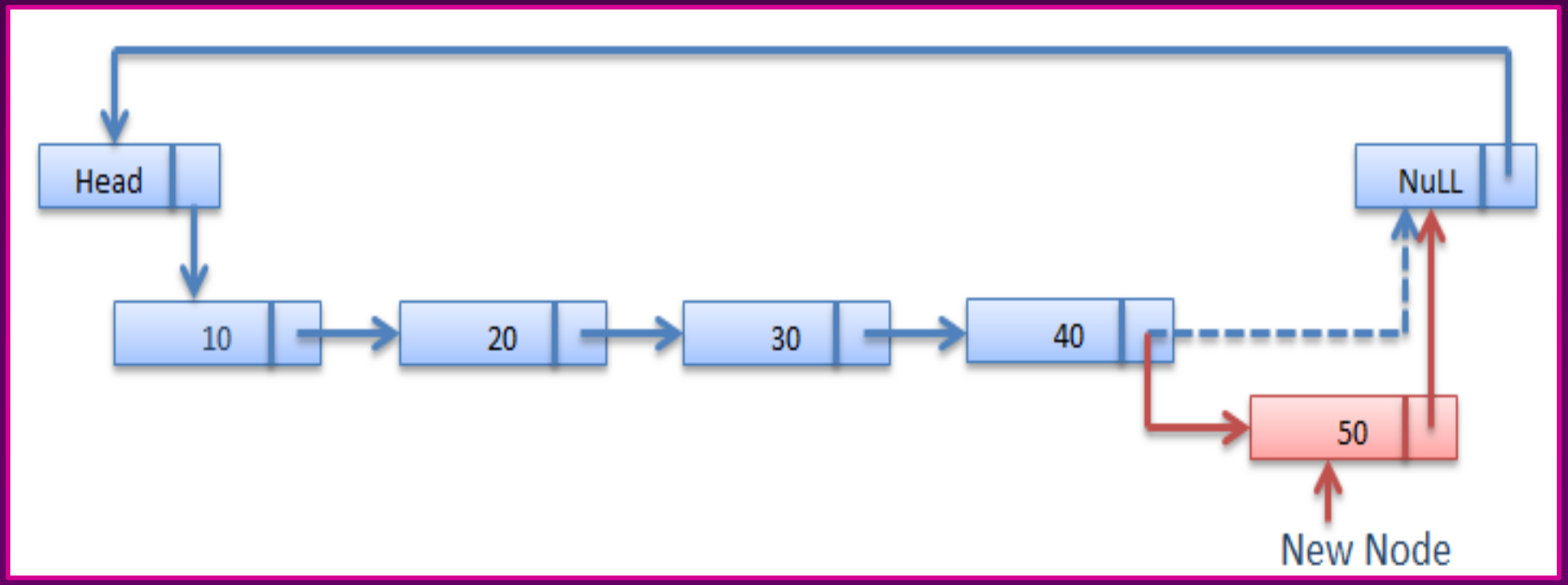
Linked List



Circular Linked List




Head
NuLL
20
30
40
50
10
New Node
Current node


Head
NuLL
10
20
30
40
50
New Node