

SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University

Department of Artificial Intelligence and Data Science

Course Name: 23ITB201 Data structures and Algorithms

II Year / III semester

Unit I – List ADTs

Topic: Circularly Linked list

Circularly Linked list

linked list is basically a linear linked list that may be singly or doubly linked. The main difference is that there is no any NULL value terminating the list. In a doubly linked list, the first node points to the next node and last node points to the first node, forming a cycle.

Types:

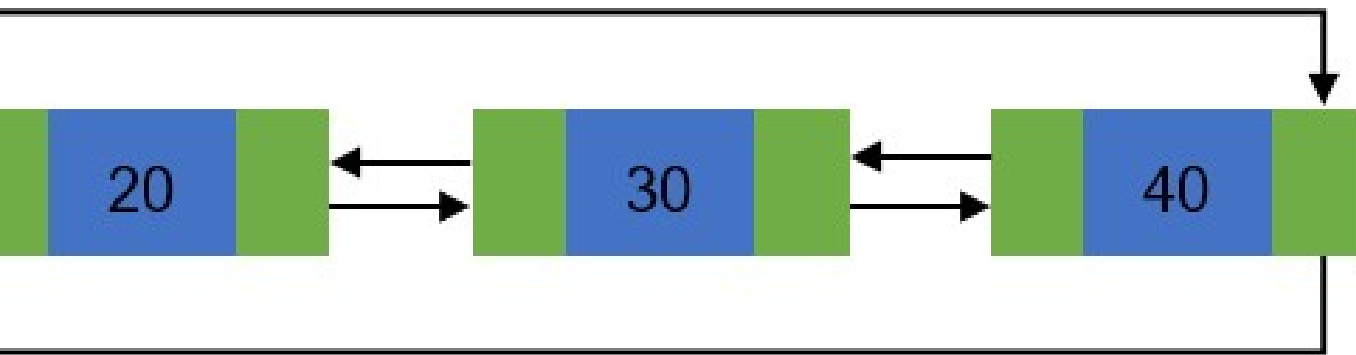
Singly linked list.

Doubly linked list.

Diagram of Circular linked list



Circular Linked List



Doubly Circular Linked List

Advantages of Circular

- Entire list can be traversed starting from any node.
- Circular lists are useful in many applications where a circular structure when traversed is required. For example, a circular queue is accessed in a circular manner.
- we can easily traverse the list in both directions. In a doubly circular linked list, each node has two pointers, one pointing to the next node and one pointing to the previous node, which is not possible in a singly linked list.

list

beginning in circular Singly linked list

```
rt_begin(int data)
```

```
de *newNode, *temp;
```

```
(struct node *)malloc(sizeof(struct node));
```

```
data = data;
```

```
next = head;
```

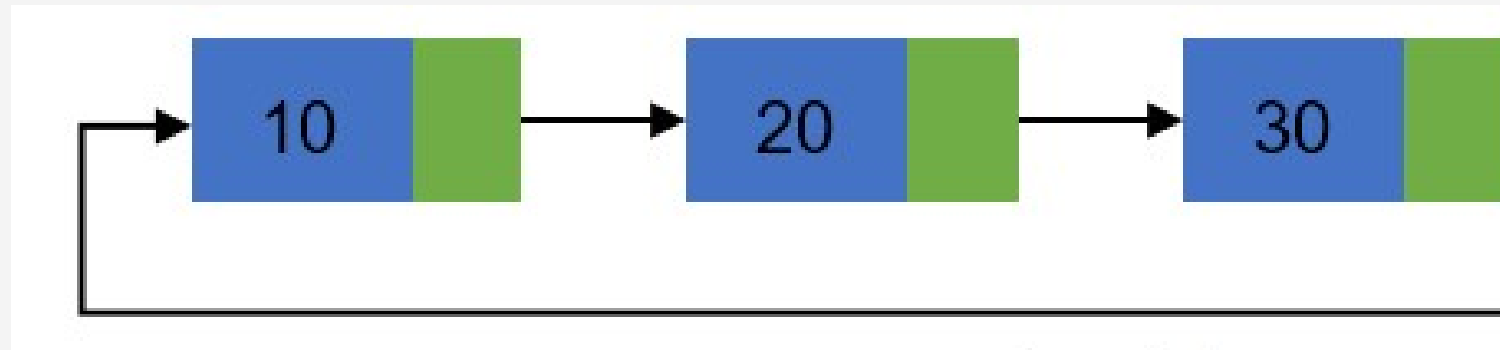
```
ad;
```

```
p->next != head)
```

```
mp->next;
```

```
t = newNode;
```

```
wNode:
```



at end in circular Singly linked list

```
ertend (int data)
```

```
ode * temp, newnode;
```

```
= (struct node *)malloc(sizeof(struct node));
```

```
>data = data;
```

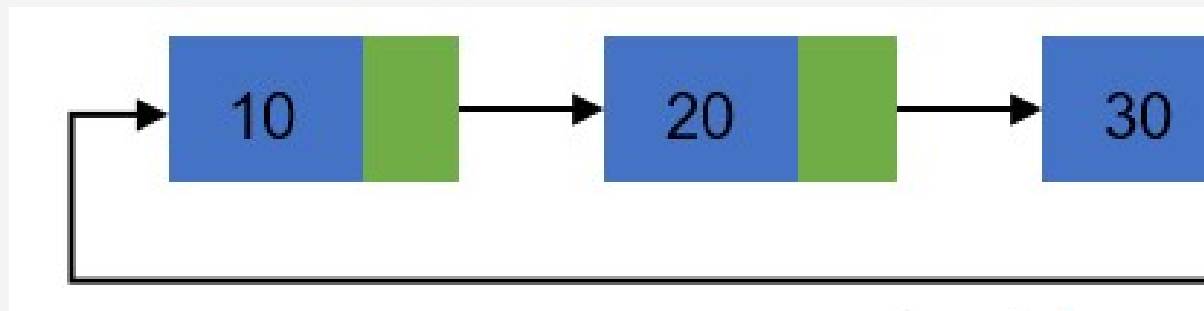
```
ead;
```

```
mp -> next != head)
```

```
emp -> next;
```

```
xt = newnode;
```

```
->next =head;
```



delete first node- CLL

```
deletefirst (int data)
```

```
node * temp, todelete;
```

```
temp = head;
```

```
while (temp != head;
```

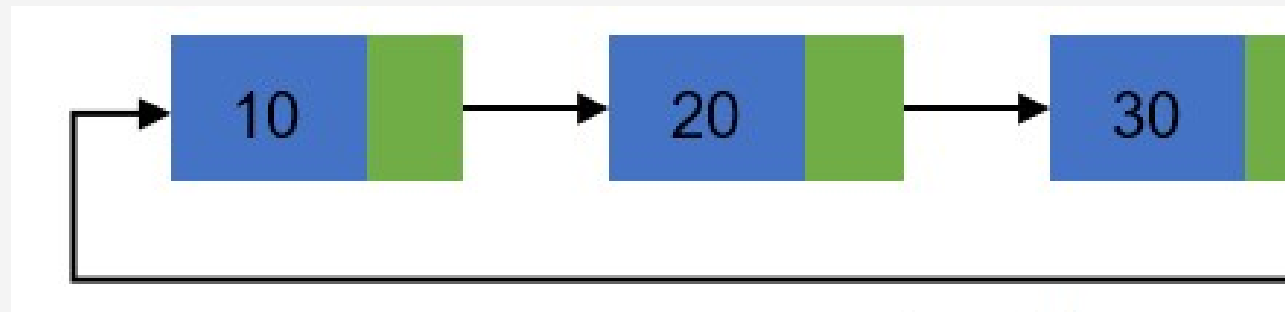
```
temp ->next != head)
```

```
temp -> next;
```

```
temp = temp -> next;
```

```
temp -> next = head;
```

```
delete todelete);
```



Displaying the content of the list- CLL

```
displayList()
```

```
struct node *temp;
```

```
temp = head;
```

```
do {
```

```
    printf("%d" , temp->data);
```

```
    temp = temp->next;
```

```
}while(temp != head);
```


h operation

```
search(struct node *head, int key)
```

```
struct node *current = head;
```

```
if (current == NULL)
```

```
return;
```

```
if (current->data == key)
```

```
return index;
```

```
current = current->next;
```

```
} while (current != head);
```

beginning

ly linked list -insertion at beginning

```
first()
```

```
le *newnode;
```

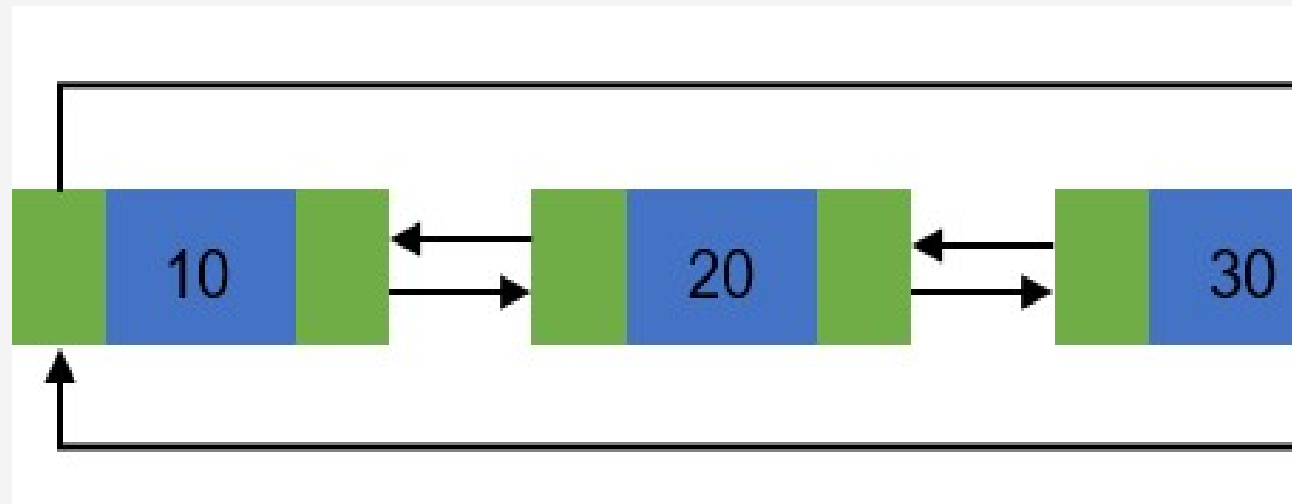
```
struct node *) (malloc(sizeof(struct
```

```
y = newnode;
```

```
/node;
```

```
y = last;
```

```
t = head;
```



used for performing polynomial operations like addition,
multiplication etc.,

structure can be implemented using linked list

structure can be implemented using linked list.

Difference between Circularly Linked list and Doubly linked list?

Advantages of Circularly linked list.

