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, Chennai

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

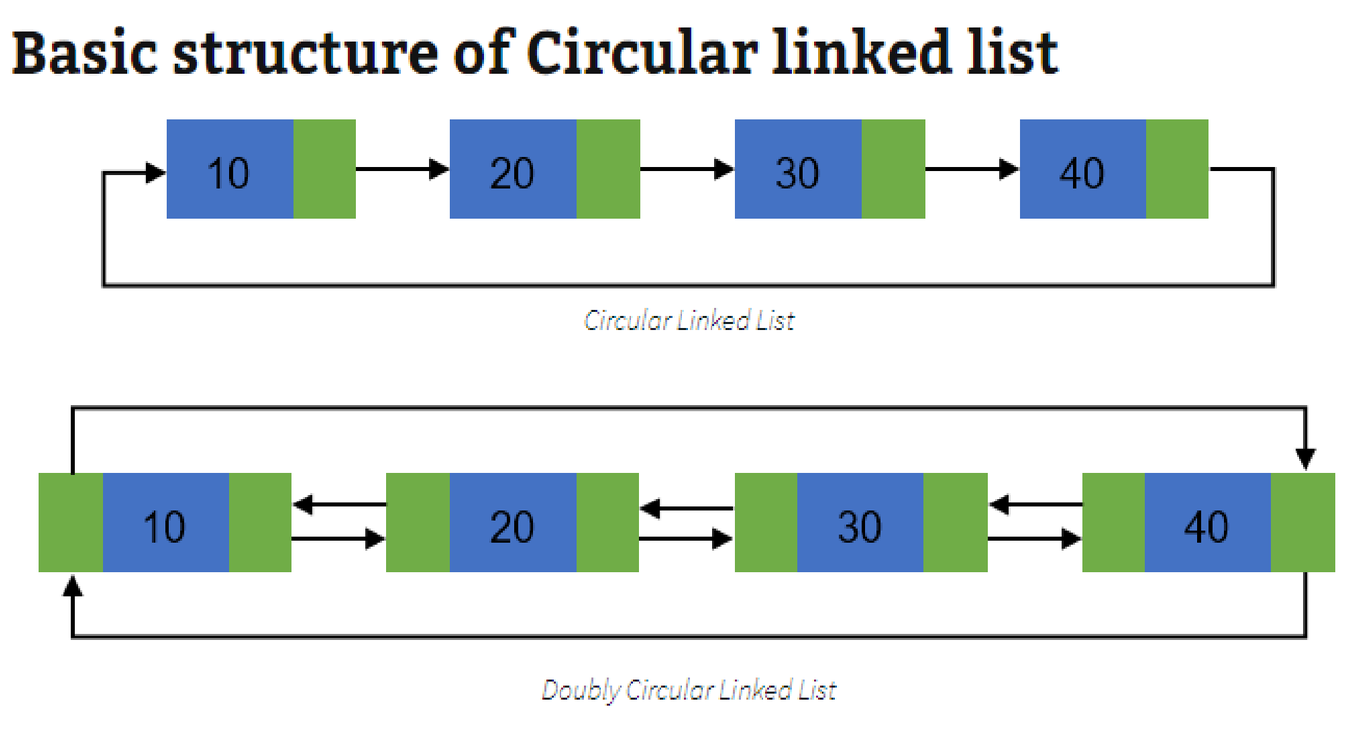
Circularly Linked list

- A circular linked list is basically a linear linked list that may be singly or doubly.
- The only difference is that there is no any NULL value terminating the list.
- Every node points to the next node and last node points to the first node, thus forming a circle.

Two variations:

- Circular Singly linked list.
- Circular doubly linked list.

Circular linked list



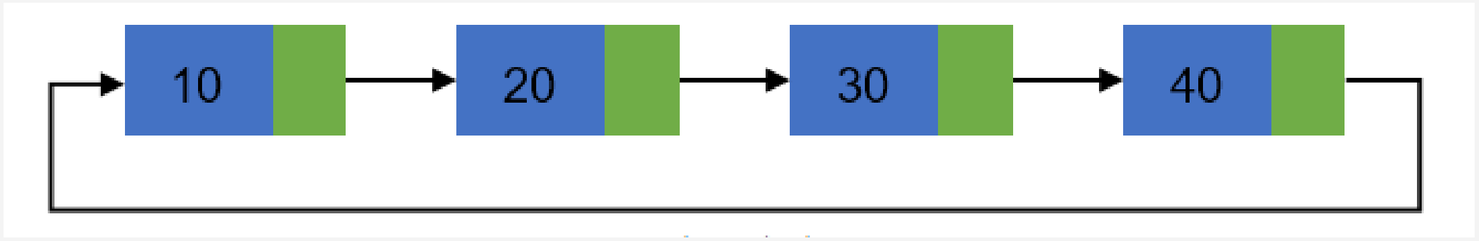
Advantages of Circular linked list

- Entire list can be traversed from any node.
- Circular lists are the required data structure when we want a list to be accessed in a circle or loop.
- we can easily traverse to its previous node, which is not possible in singly linked list.

Insert at beginning in circular singly linked list

Insert at beginning in circular Singly linked list

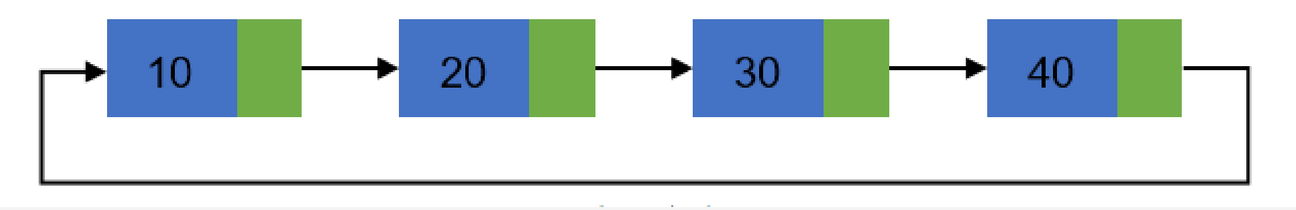
```
void insert_begin(int data)
{
struct node *newNode, *temp;
newNode = (struct node *)malloc(sizeof(struct node));
newNode->data = data;
newNode->next = head;
temp = head;
while(temp->next != head)
{
temp = temp->next;
}
temp->next = newNode;
head = newNode;
}
```



Insert at end in circular singly linked list

Insert at end in circular Singly linked list

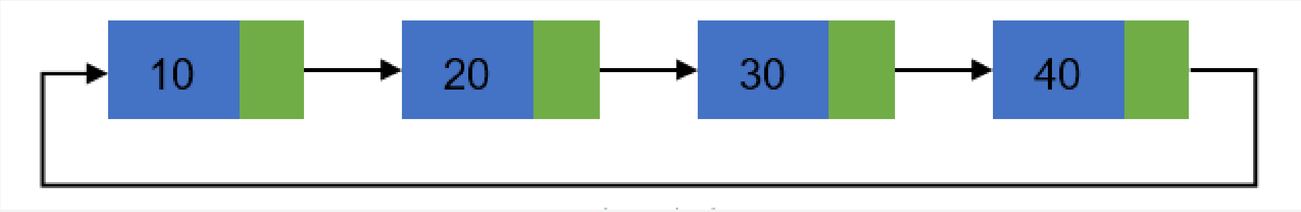
```
void insertend (int data)
{
struct node * temp, newnode;
newNode = (struct node *)malloc(sizeof(struct node));
newNode->data = data;
temp = head;
while(temp -> next != head)
{
temp = temp -> next;
}
temp->next = newnode;
newnode ->next =head;
}
```



Delete first node- CLL

Delete first node- CLL

```
void deletefirst (int data)
{
struct node * temp, todelete;
todelete = head;
temp = head;
while(temp ->next != head)
{
temp =temp -> next;
}
head = head -> next;
temp -> next =head;
free(todelete);
}
```



Displaying the content of the list- CLL

Displaying the content of the list- CLL

```
void displayList()
{
    struct node *temp;
        temp = head;
        do {
            printf("%d" , temp->data);
            temp = temp->next;

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

Search operation

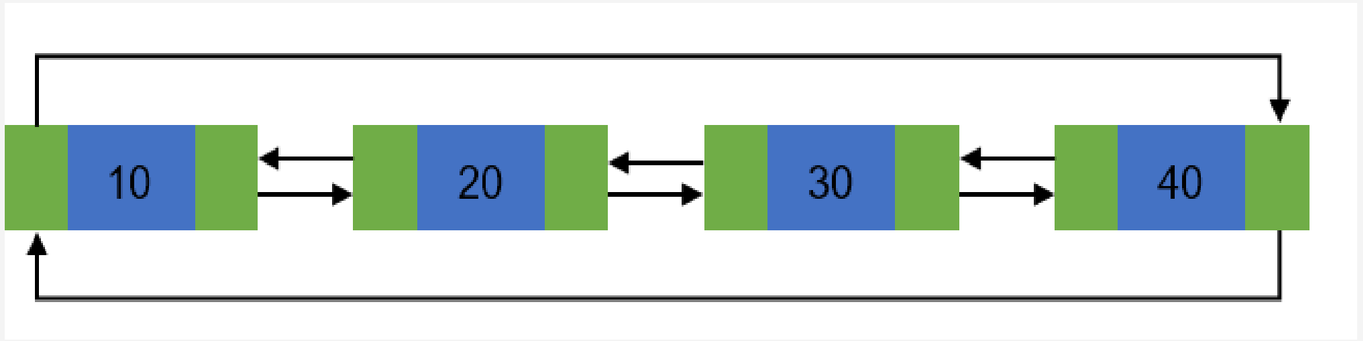
Search operation

```
int search(struct node *head, int key)
{
    struct node *current = head;
    do
    {
      if (current == NULL)
 return;
if (current->data == key)
                    return index;
  current = current->next;
      } while (current != head);
}
```

Circular Doubly linked list – insertion at beginning

Circular Doubly linked list –insertion at beginning

```
void insert_at_first()
{
    struct node *newnode;
   newnode = (struct node *)(malloc(sizeof(struct
node)));
    head->prev = newnode;
    head = newnode;
    head->prev = last;
    last->next = head;
}
```



Applications of Linked List

- Linked list is used for performing polynomial operations like addition, subtraction, multiplication etc.,
- Stack data structure can be implemented using linked list
- Queue data structure can be implemented using linked list.
- Radix sort
- Multi list

Assessment

- Difference between Circularly Linked list and Doubly linked list?
- Advantages of Circularly linked list.


thank
you