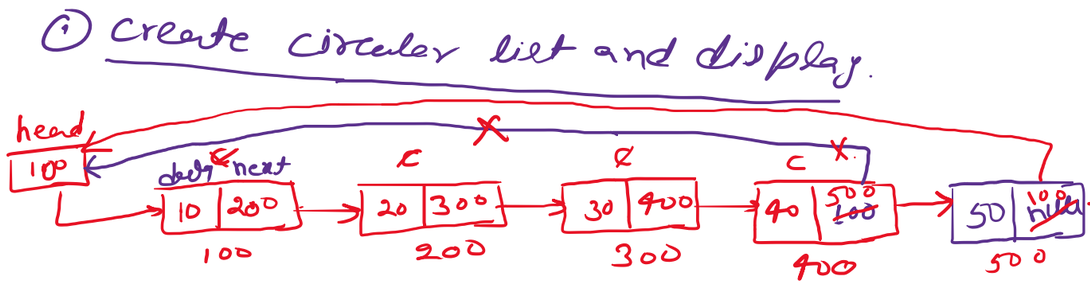
① circular linked list
② Recursion
③ sorting
④ Stack
⑤ Queue

① Bubble Sort
② Insertion Sort
③ Selection Sort
④ Merge. Sort
⑤ heap sort
⑥ Quick Sort

Circular Linked List =>

① ~~ins~~ Creat list and display list
② insert beginning, end, between
③ Delete beginning, end, between.
④ Reverse circular linked list

① Create circular list and display.

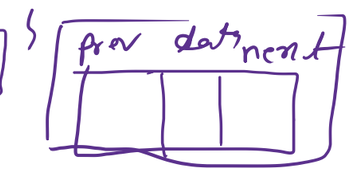


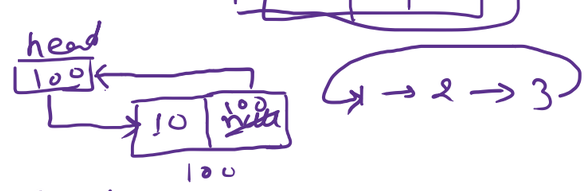

```
Node{
int dat;
Node next;
Node prev;
```

```
public void createCircular(int data){
    Node n1 = new Node(data);
    if(head == null){
        head = n1;
        n1.next = head;
    }else{
        Node current = head;
        while(current.next != head){
            current = current.next;
        }
        current.next = n1;
        n1.next = head;
    }
}
```

c.next != null
c.next ! head

C1=100

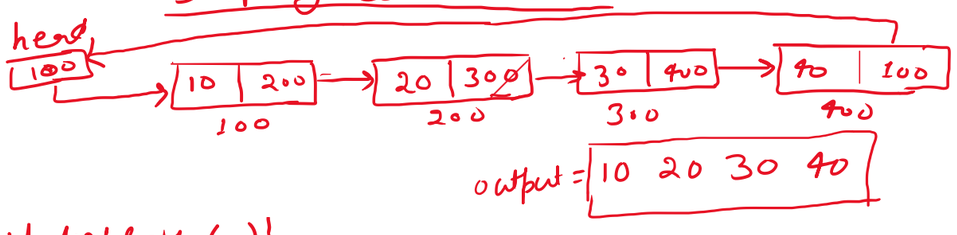




Circular
① circular singly Linked list
② Circular Doubly Linked list

C=400

Display Circular List



output = 10 20 30 40

C1 = current

C = ~~500~~ 400 300 200 100 C1 = head

while(C1 != head){

100 != 100

100 => 100

C1) 200 != 100

```
void display(){
    if(head != null){
        Node c1 = head;
        do{
            Sop(c1.data);
            c1 = c1.next;
        }while(c1 != head);
    }
```

Insert Beginning of the circular list


```
void insertBeginning(int data){
    if(head != null){
        Node cr = head;
        while(cr.next != head){
            cr = cr.next;
        }
        Node n1 = new Nod(data);
        n1.next = head;
        head = n1;
        cr.next = head;
    }
```

16-07-2023 covered topic
① create circular list
② display circular list
③ insert beginning of the list

17-07-2003 topic
① insert last of the circular list
② insert between of the circular list
③ delete operation (beginning, last, between)