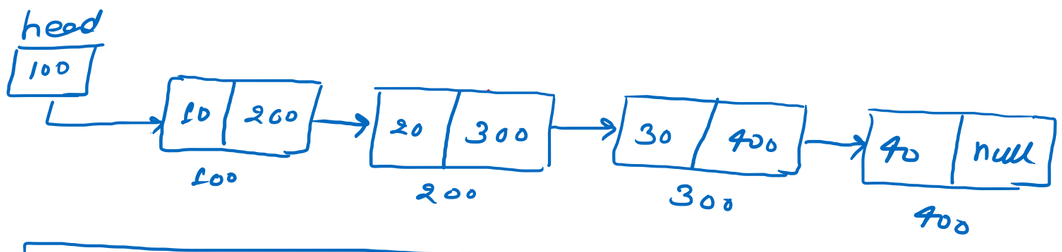
Queue with linked list



rear → enqueue (insertion)
front → dequeue (deletion)

```
Class LinkedList{
  Node head
  Class Node{
    int data;
    Node next;
    Node(int data){
      this.data = data;
    }
```

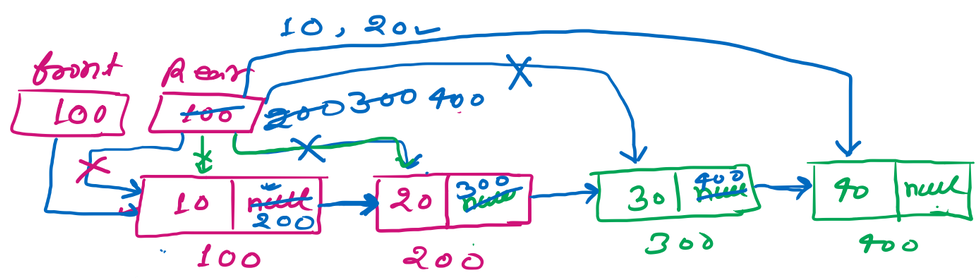
structure of the linked list.

```
Class QueueWithLinkedList{
  Nod front, rear;

  Class Node{
    int data;
    Node next;
    Node(int data){
      this.data = data;
    }
}
```

front → null
rear → null

```
public void enqueue(int data){
    Node newNode = new Node(data);
    if(rear == null){
      this.rear = this.front = newNode
    }else{
        this.rear.next = newNode;
        this.rear = newNode;
    }
}
```

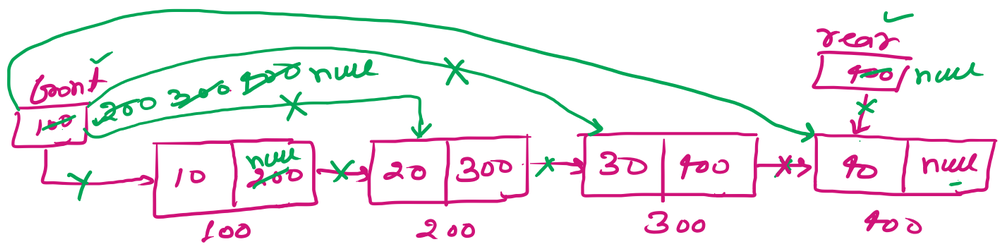


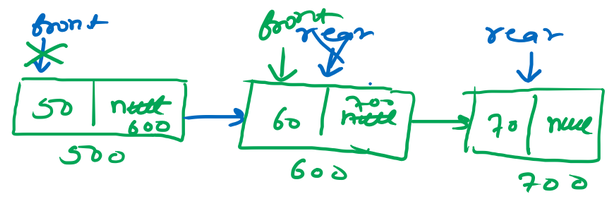
f = 0 1
r = 0 1 2 3 4 5 6
r = 6-1

10	20	30	40	50	60
0	1	2	3	4	5
1st	2nd	3rd	4th	5th	6th

queue[front] ⇒ 10 1st
front++ ⇒ 20 → 2nd

this.rear.next = newNode;
200.next = 300
this.rear = newNode;





temp = 100 200 300 400
rear = 400 null 500 600 700
front = 400 null 500
temp = 400 500

```
public void dequeue(){
    if(this.front != null){
      Node temp = this.front;
      this.front = this.front.next;
      temp.next = null;
      Sop("Dequeue data 3" + temp.data); // 10 20, 30 40
    {if(this.front == null){
        this.rear = null;
    }
}else{
    Sop("Queue is underflow");
    }
}
```