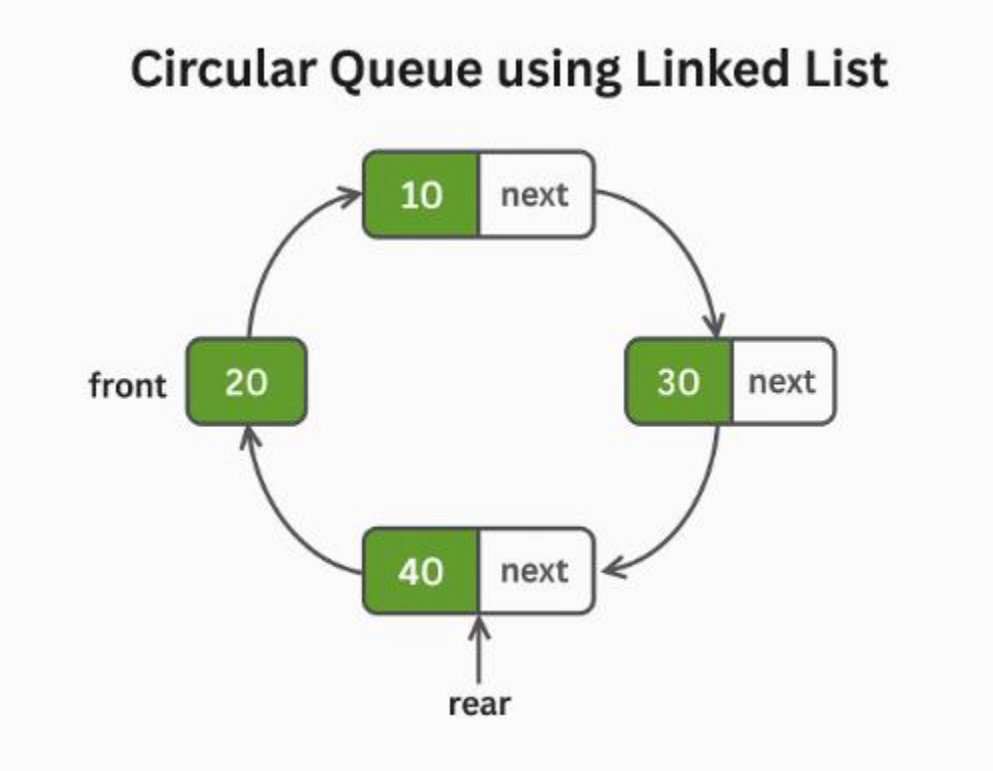
Circular Queue Using Linked List



Think of a **circular queue** like people standing in a **round circle**.

Each person is holding the **hand of the next person**.

Important points:

1. Every person is a Node.

A node has:

- **some data (like a number)**
- **a pointer (which tells who is next)**

2. The last person does NOT point to NULL.

Instead…
The last person points back to the first person
This is what makes it **circular**.

3. Two pointers are used:

- **front** → first person in the circle
- **rear** → last person in the circle

4. Enqueue (Add an element):

We add a new person **next to the rear**.

5. Dequeue (Remove an element):

We remove the **front person**, and now the next person becomes the front.

6. Peek:

Peek means:
"Tell me who is standing at the front."

Example

Imagine friends standing in a circle:

20 → 10 → 30 → 40 → back to 20

- **20 is front**
- **40 is rear**
- 40 points back to 10 (circular)

Why Circular?

Because once rear reaches the end,
you don’t have to start again
 you continue from the beginning

Just like a round walking track.