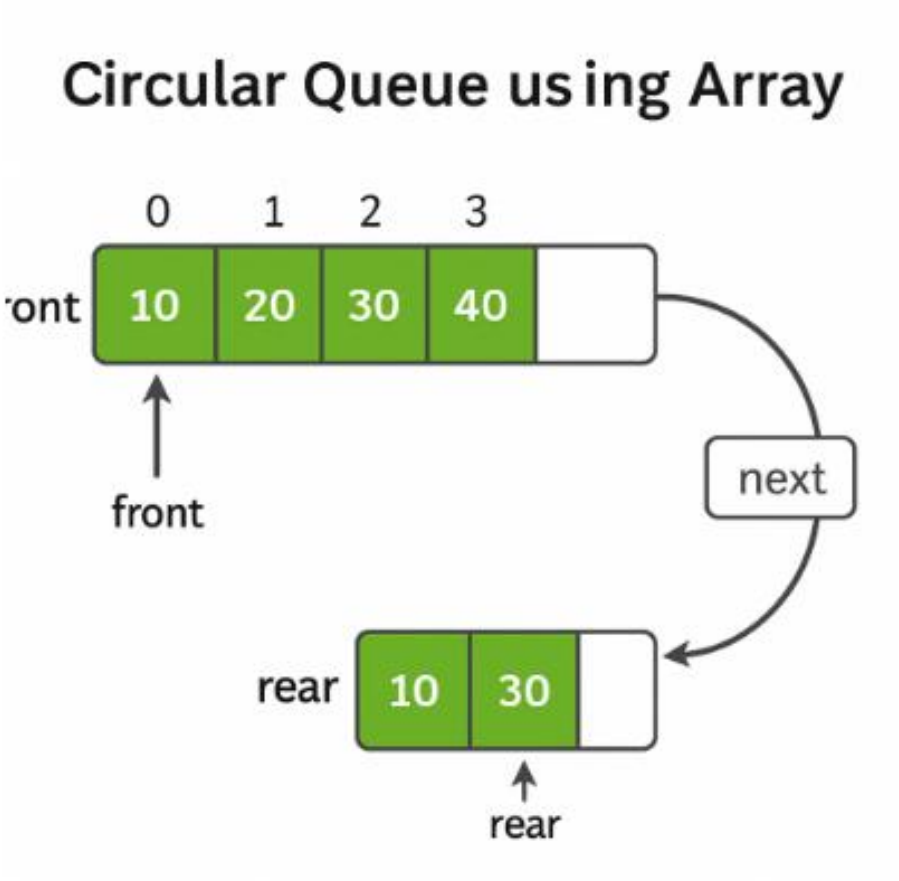

What the diagram shows

The diagram represents a **circular queue** stored inside an **array**.

Example array contents:

Index: 0 1 2 3 4
Value: 10 20 30 40 (empty)

front = 0

(pointing to 10)

rear = 3

(pointing to 40)

Why is it called "circular"?

Because when you reach the end of the array **(index 4),** the next position wraps around to index **0**.

This is done using:

rear = (rear + 1) % capacity

So the queue behaves like a circle even though it's stored in a straight array.

How enqueue works

When you do:

enqueue(10)
enqueue(20)
enqueue(30)
enqueue(40)

Array looks like this:

```
+------+------+------+------+------+
| 10   | 20   | 30   | 40   |      |
+------+------+------+------+------+
  ↑                    ↑
front = 0              rear = 3
```

If you add one more element:

rear = (3 + 1) % 5 = 4

So the next element goes at index **4**.

If you add another:

rear = (4 + 1) % 5 = 0

THIS is the circular part!

How dequeue works

When you remove:

dequeue()

The element at front (10) is removed.

Then:

front = (front + 1) % capacity
front = (0 + 1) % 5 = 1

Now front points to 20.

Why this is useful?

A circular queue:

- avoids memory waste
- reuses empty positions
- works fast (O(1) operations)
- is perfect for buffers, scheduling, and real-time systems