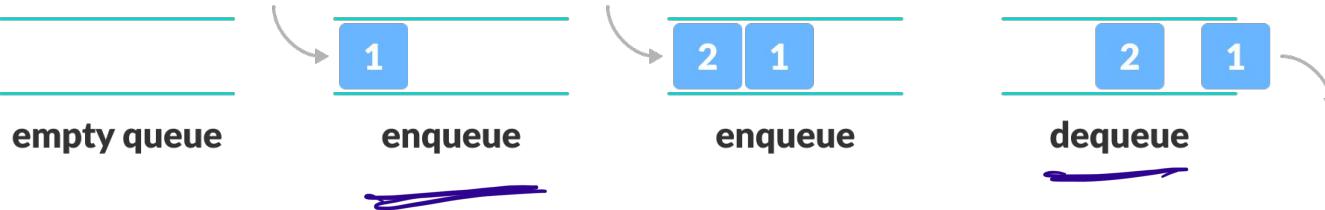
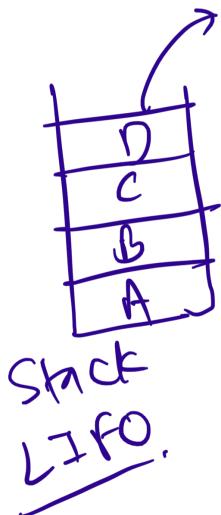
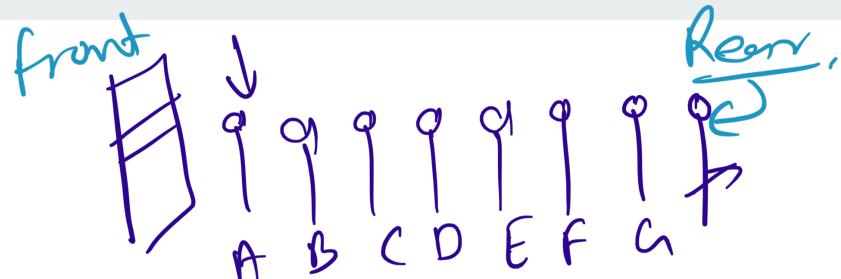


Queue Basics



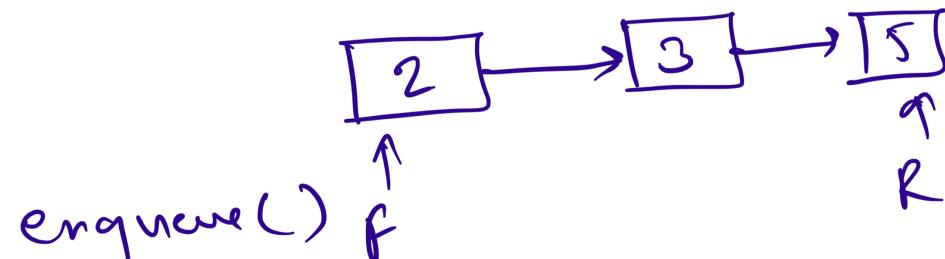
Queue Data Structure



Implement Queue Using Linked List

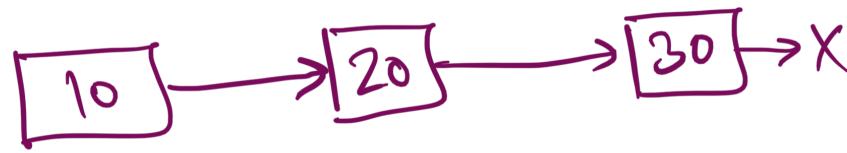
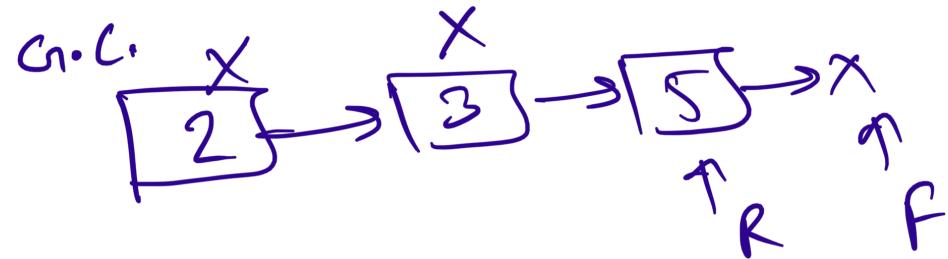
```
boolean isEmpty(){  
    return front == null;  
}
```

front = rear = null

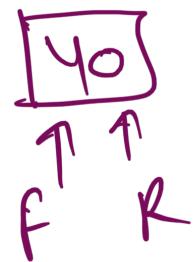


(Insert at End of
LL)

dequeue ()

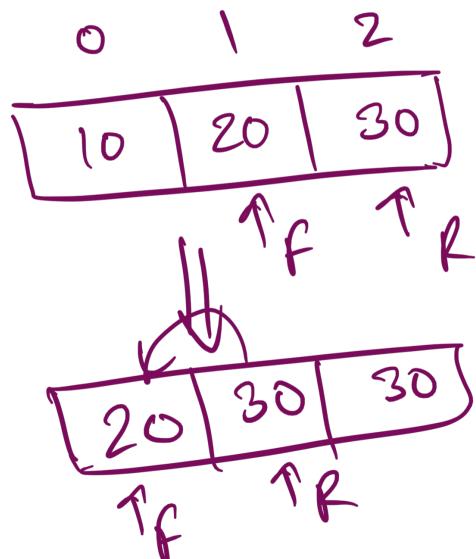


$$\text{ans} = \underline{\underline{30}}$$



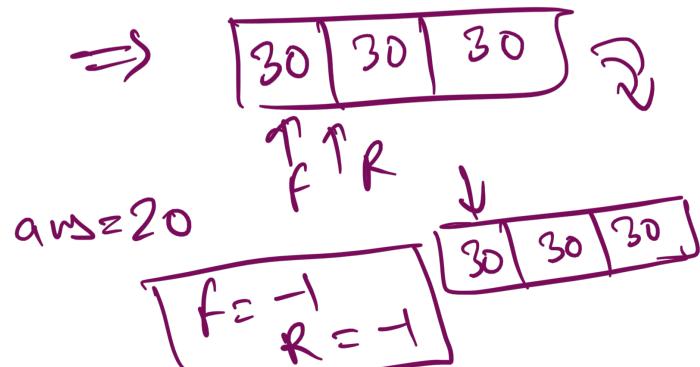
Implement Queue Using Array

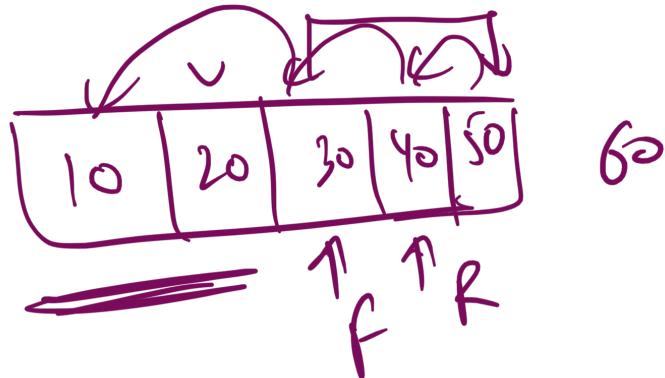
int[] =



Overflow.

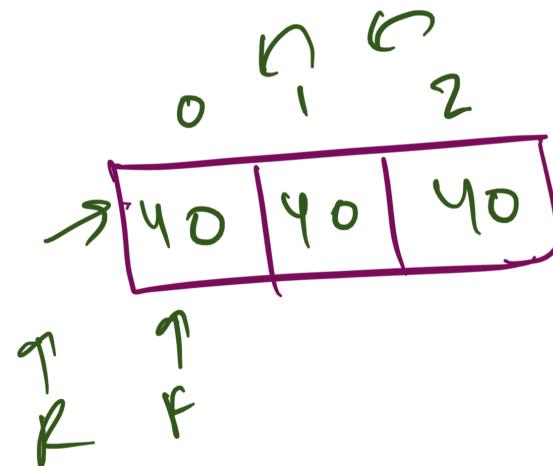
$$f = 0 \\ R = \phi \times 2$$





Space Not
occupied property
hence slide to left

// f always
0

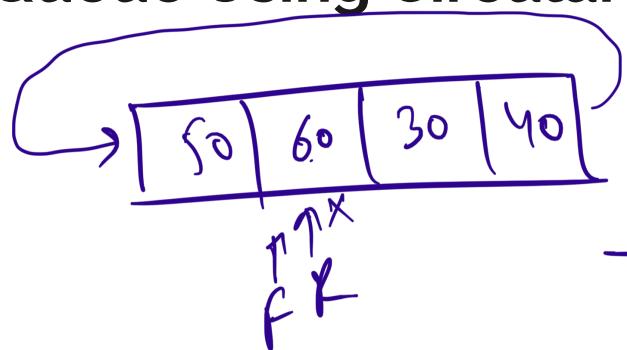


$$\begin{array}{l} f = 0 \\ R = \emptyset \times \emptyset \times \emptyset \end{array}$$

ans = 40

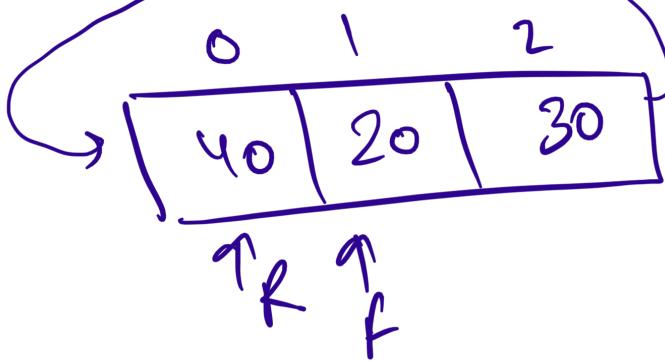
Implement Queue Using Circular Array

% Modulo



fIFO

$f = R = -1$



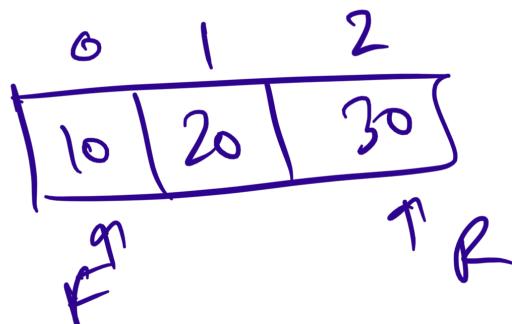
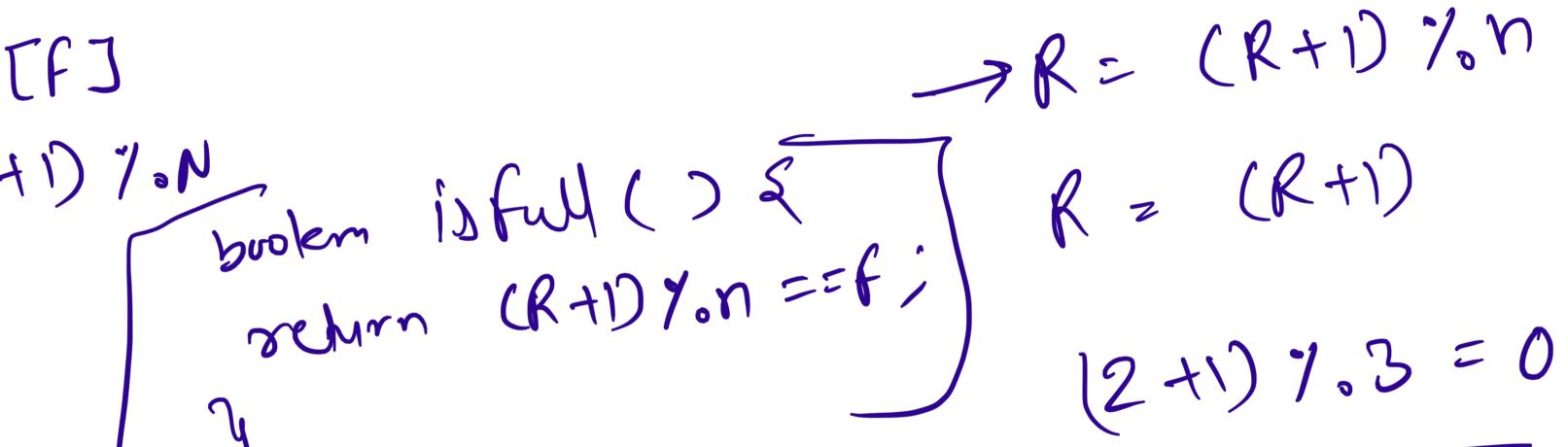
$$n = 3$$

$$f = 1$$

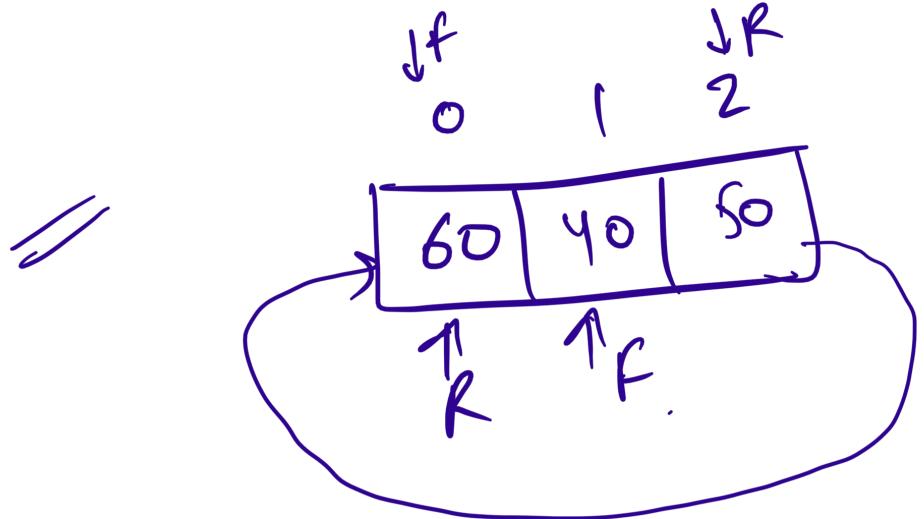
$$R = x \neq 0$$

ans $\leftarrow a[f]$

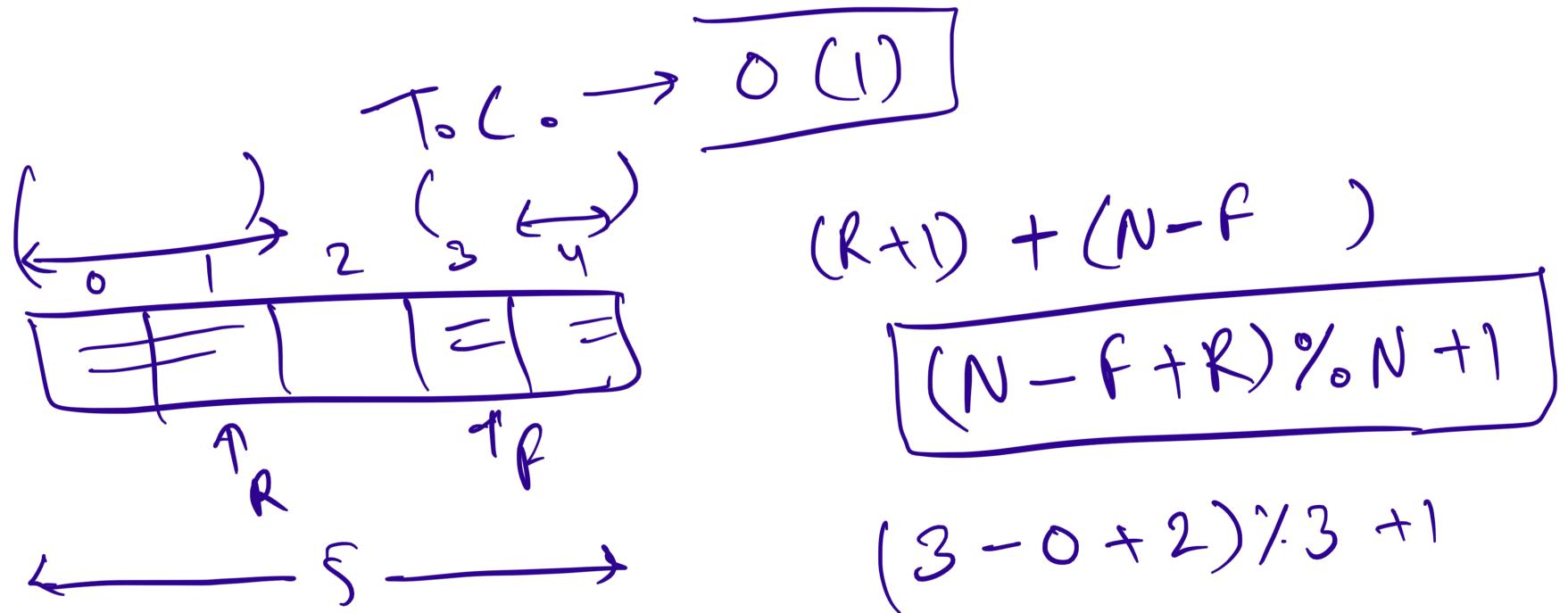
$$f = (f + 1) \% n$$



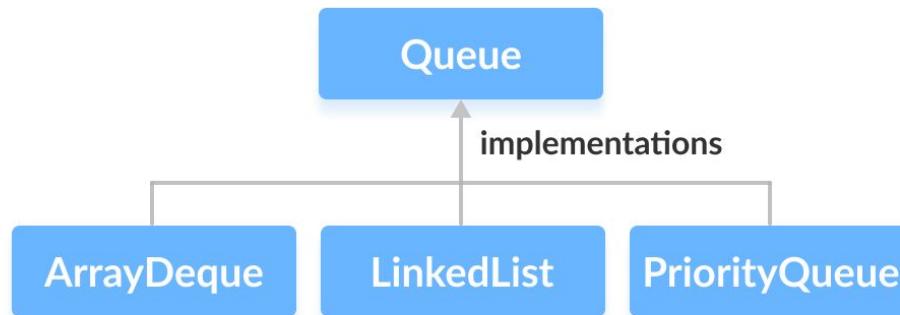
$f = -1 \uparrow ?$
empty $\uparrow ?$



$$\begin{aligned}
 f &= \emptyset \neq 0 \\
 R &= -\infty \neq -\infty \neq 0 \\
 \text{ans} &= 30
 \end{aligned}$$



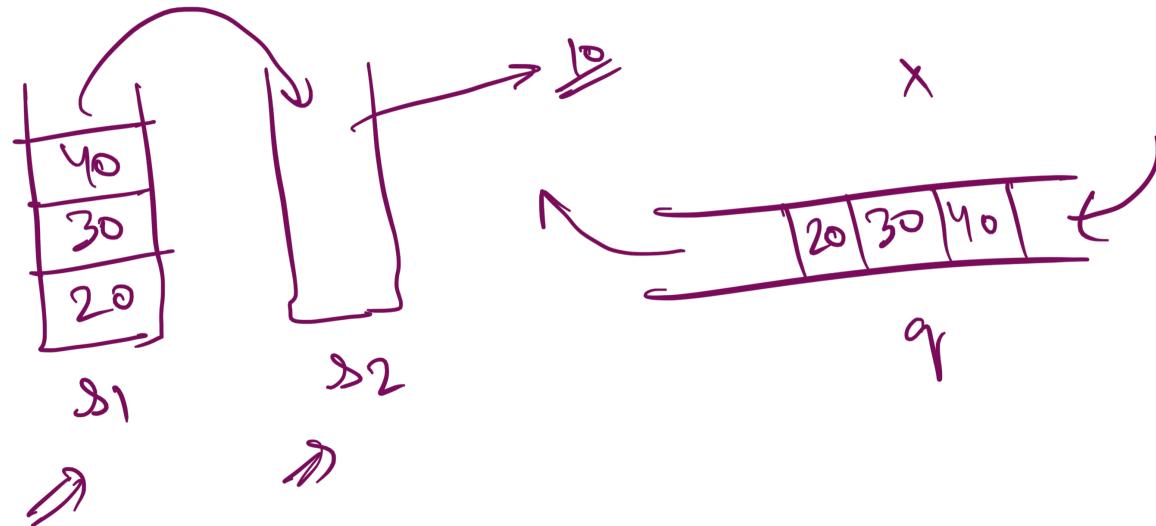
Queue Interface in Collection Framework



Methods of Queue

- **add()** - Inserts the specified element into the queue. If the task is successful, `add()` returns `true`, if not it throws an exception.
- ✓ • **offer()** - Inserts the specified element into the queue. If the task is successful, `offer()` returns `true`, if not it returns `false`.
- **element()** - Returns the head of the queue. Throws an exception if the queue is empty.
- ✓ • **peek()** - Returns the head of the queue. Returns `null` if the queue is empty.
- **remove()** - Returns and removes the head of the queue. Throws an exception if the queue is empty.
- ✓ • **poll()** - Returns and removes the head of the queue. Returns `null` if the queue is empty.

Q. Implement Queue using Two Stacks



Practice Problems

1. Implement Stack using Two Queues
2. Implement Stack using One Queue
3. Reverse a Queue
4. Reverse the first k elements of the Queue