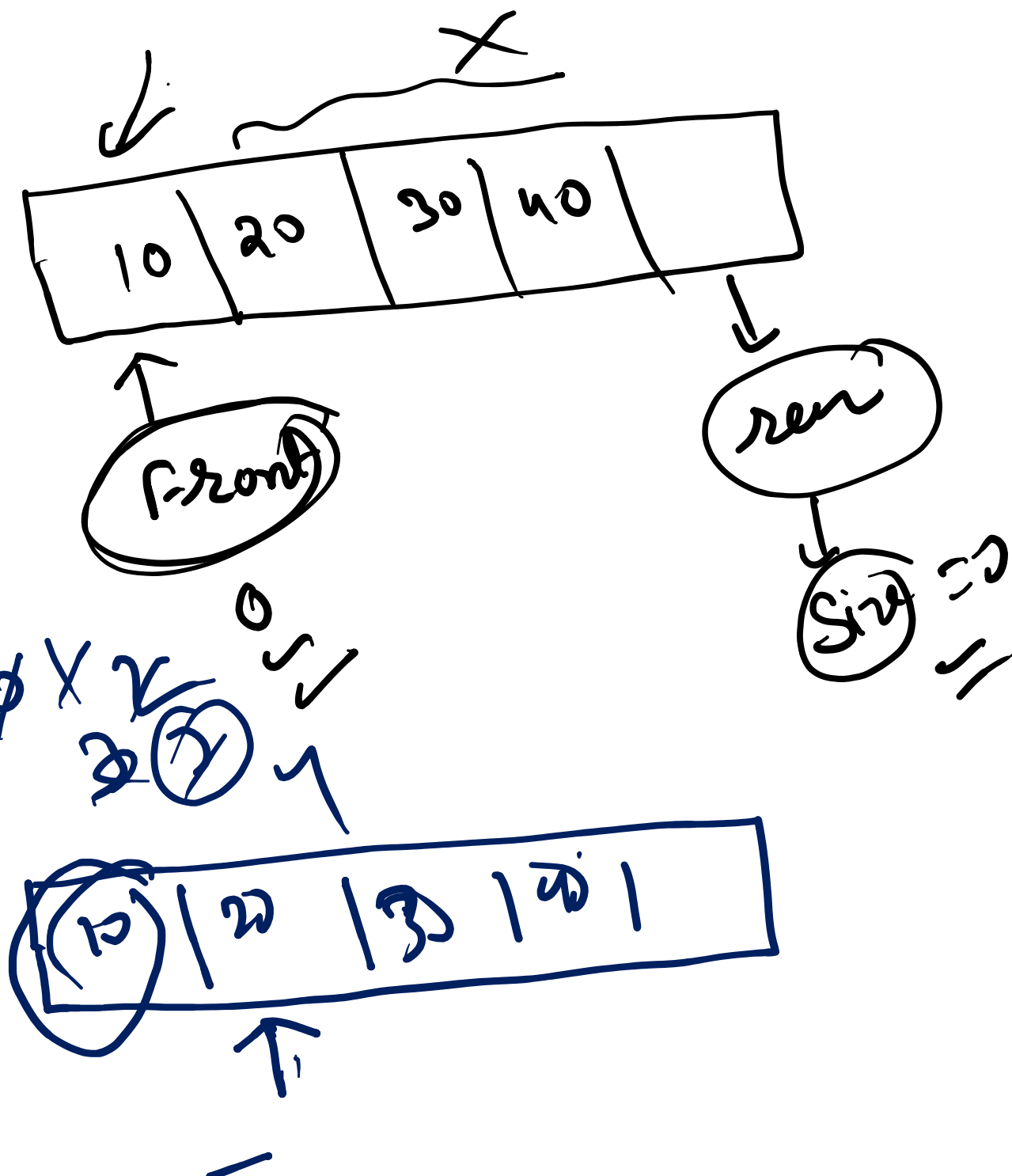


Queue → Linear / Circular
FIFO

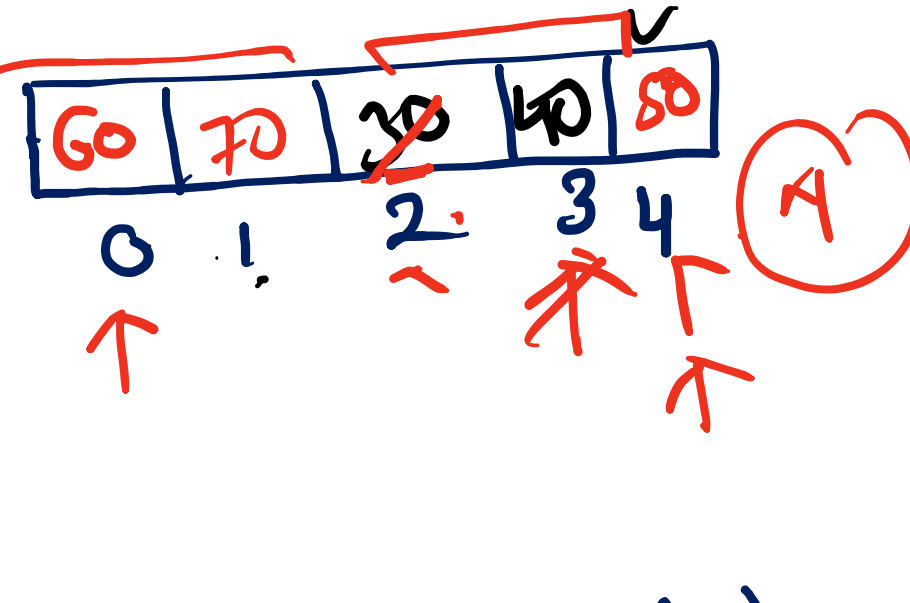


eq(10) ✓
eq(20) ✓
eq(30) ✓
eq(40) ✓

enqueue
dequeue
size

enqueue (idx item) ✓
arr[size] = item
size++

```
public void Enqueue(int item) throws Exception {  
    if (isFull()) {  
        throw new Exception("Bklol Queue full hai");  
    }  
    int idx = front + size;  
    arr[idx] = item;  
    size++;  
}  
  
public int Dequeue() throws Exception {  
    if (isEmpty()) {  
        throw new Exception("Bklol Queue Empty hai");  
    }  
    int v = arr[front];  
    front++;  
    size--;  
    return v;  
}
```



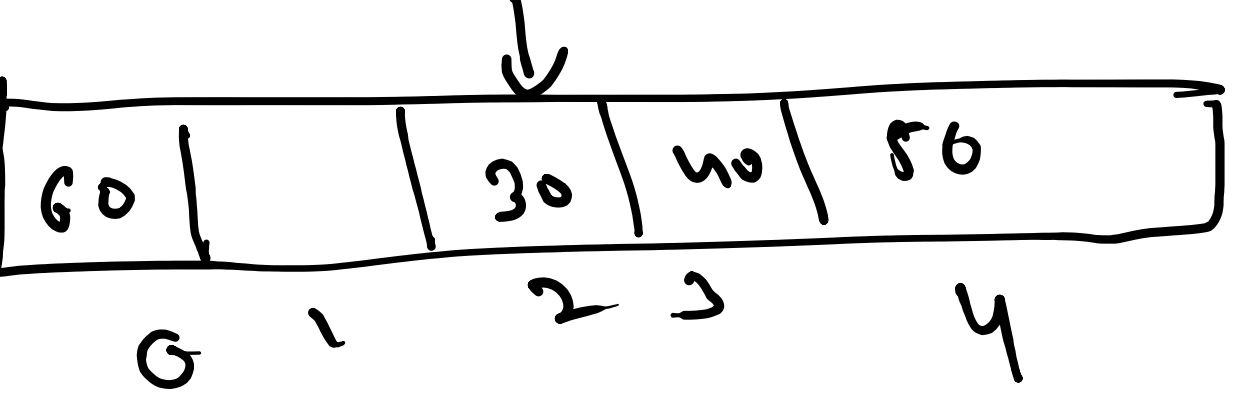
$2+3=5 \div 5=0$
 $(2+4)=6 \div 5=1$

dr
dr
dr

eq(10) ✓
eq(20) ✓
eq(30) ✓
eq(40) ✓
deq() ✓
deq() ✓
eq(50) ✓
eq(60) ✓
eq(70) ✓

Size = 4

10 P+0 = 20 = 2 ✓
20 P+1 = 21 = 3 ✓
30 P+2 = 22 = 4 ✓
40 P+3 = 23 = 5 ✓
(P+4) / arr.length



30 40 50 60