

# DSA through C++

## Assignment-11

### Doubt Class



Saurabh Shukla (MySirG)

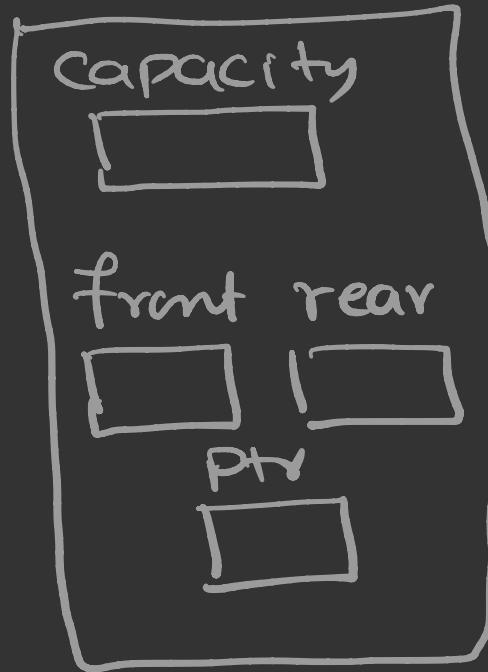
## Queue using Arrays

Class Queue

{

```
int capacity;  
int front, rear;  
int *ptr;
```

}

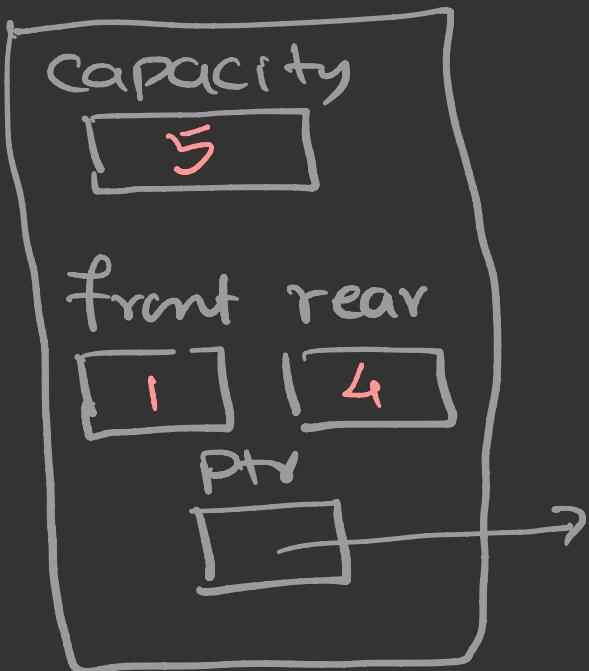


## isfull()



- $\text{front} == 0 \& \& \text{rear} == \text{capacity} - 1$
- $\text{rear} + 1 == \text{front}$

# Insert

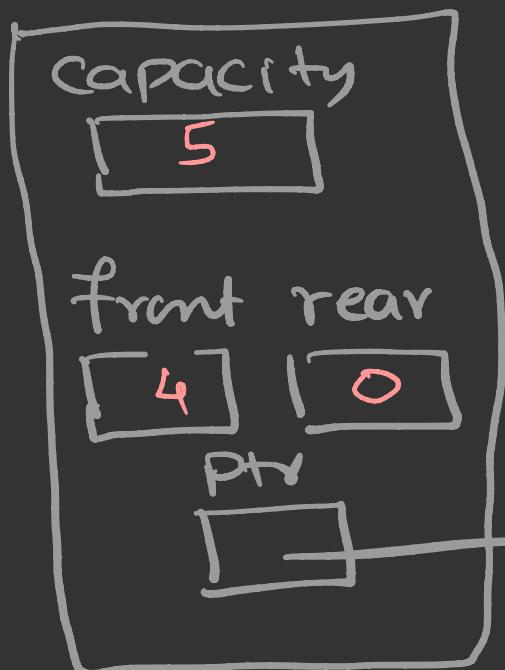


- 1) Queue empty  
front == -1  
rear == -1
- 2) Normal  
rear++
- 3) rear == capacity - 1  
rear = 0

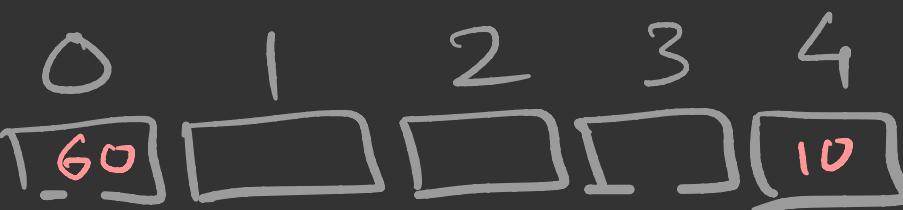
# Delete

① Queue empty

deletion not possible



- ①
- ③
- ④
- ②



2) Normal

front++

single element  
3) front == rear  $\neq -1$

front = -1

rear = -1

front = 0

① Queue empty

deletion not possible

① Queue empty

deletion not possible

# Deep Copy

Q



Current Object



Queue  $Q_1(5)$ ;

...

Queue  $Q_2 = Q_1$ ;

Queue  $Q_3(10)$ ;

...

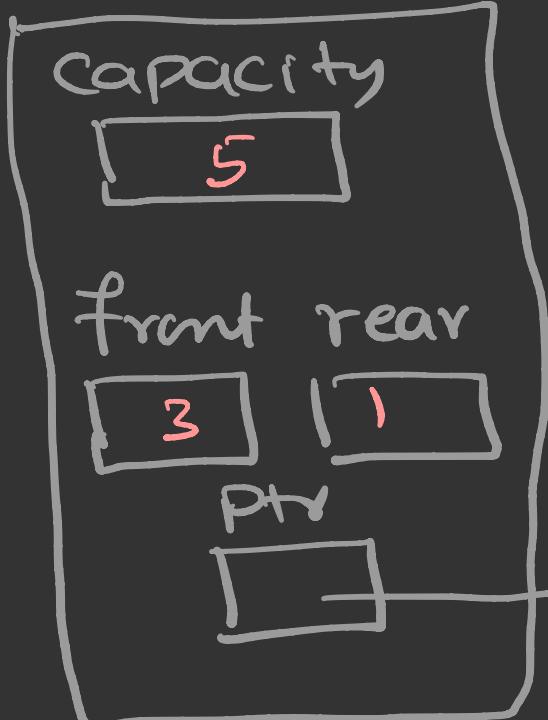
- - -

$Q_3 = Q_1$ ;

```

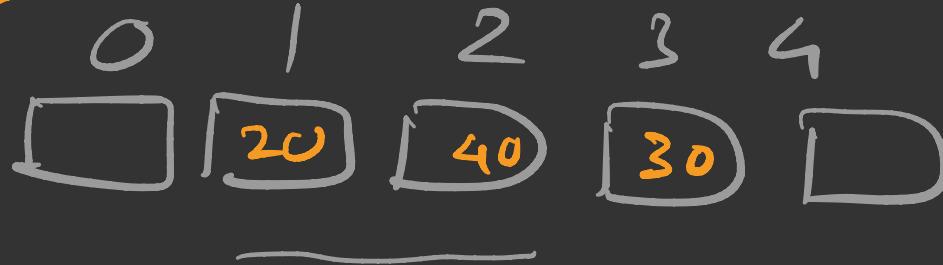
i = Q.front;
ptr[i] = Q.ptr[i];
if (i == capacity - 1)
    i = 0;
else
    i++;

```

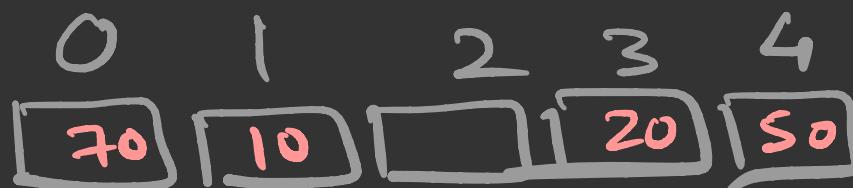


Count

$f=1$        $r=3$        $r-f+1$



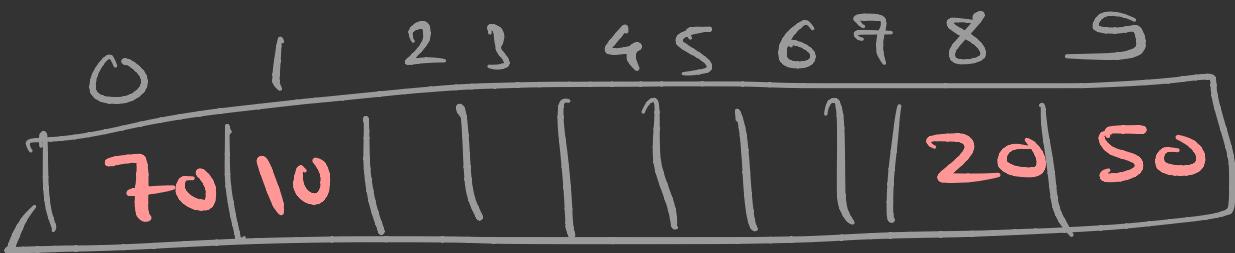
$(f < r)$



$(f > r)$

if ( front == -1)  
return 0;

```
c=0
i=front;
while( i != rear)
{
    if(i==capacity-1)
        i=0;
    else
        i++;
    c++;
}
return c+1;
```



Capacity-(front - rear - 1)

Capacity - front + rear + 1

enqueue Q1(5),  $\leftarrow$   $f=4$   
 $r=1$   
enqueue Q2(10);

0	1	2	3	4
10	20			40

Q2 = Q1;

$\leftarrow$   $f=4$   
 $r=1$

