



Queue

Agenda

- Queue
- Queue principles and operations
- Queue isFull(), isEmpty() functions
- Queue example
- Front and Rear Pointers
- Queue Insertion pseudocode
- Queue Deletion pseudocode
- Circular queue
- Circular queue example
- Circular Queue isFull(), isEmpty() functions
- Circular Queue Insertion pseudocode
- Circular Queue Deletion pseudocode

Queue

Idea:

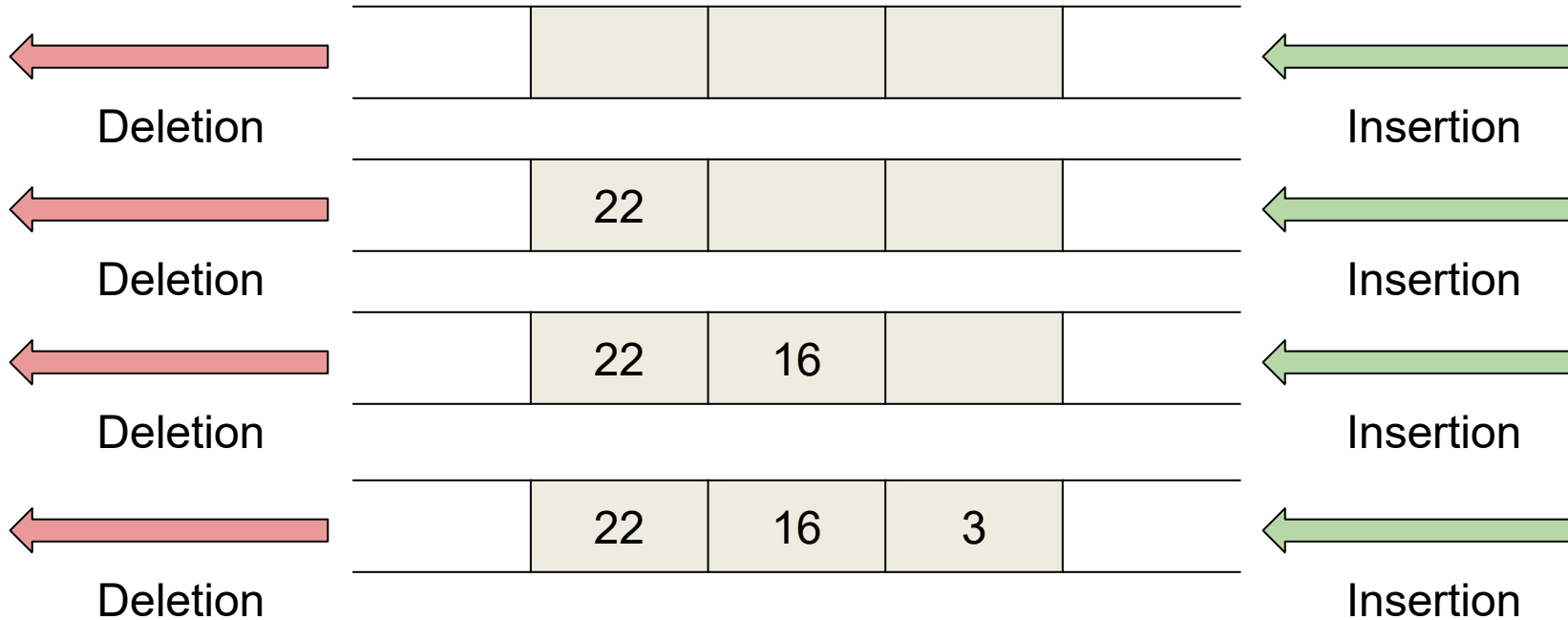
- Queue is where Insertion of elements happens from one end and deletion of elements happens from the other end of the list.
- Queues take **First In First Out Approach (FIFO)**.

Example: Vehicles near Toll Gate.

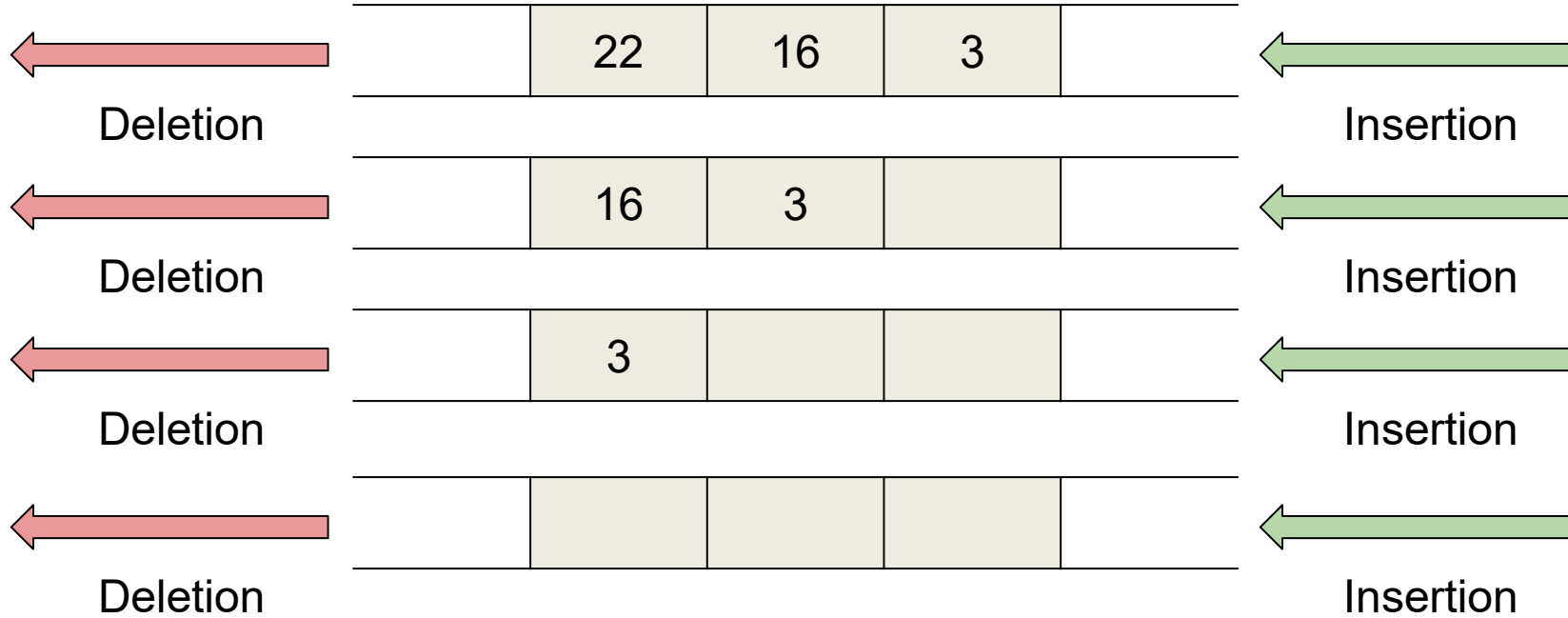


Queue insertion example

Insert 22,16,3 and empty queue



Queue deletion example



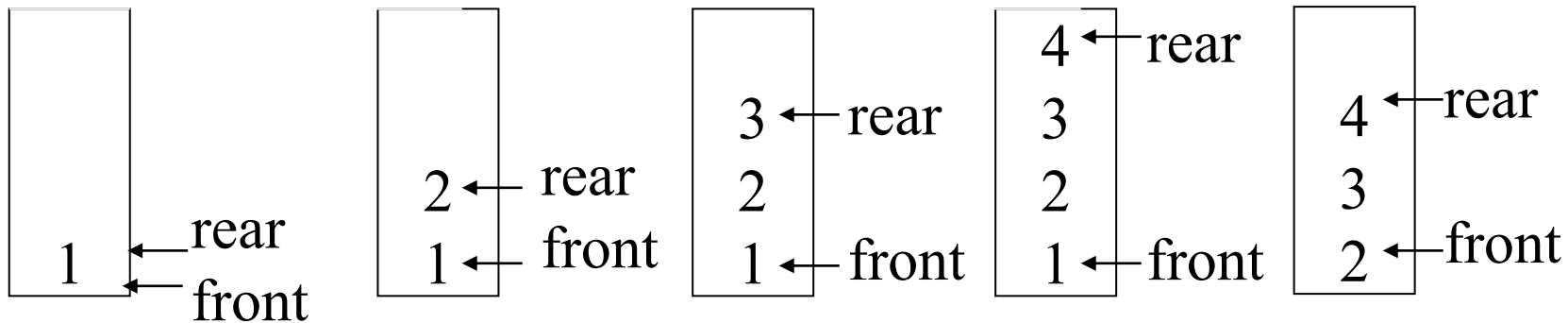
Front & Rear pointers

Front → contains address of starting element in queue

Rear → contains address of last element in queue

We perform insertion at rear and deletion at front end of queue

Front = Rear = -1



Queue isFull() , isEmpty() functions

```
public boolean isFull() {  
    if (rear == maxSize - 1) {  
        return true;  
    }  
    return false;  
}
```

```
public boolean isEmpty() {  
    if (front == -1) {  
        return true;  
    }  
    return false;  
}
```

Queue insertion method

```
public void enqueue(int data) {  
    if (!isFull()) {  
        rear++;  
        arr[rear] = data;  
        if (front == -1)  
            front = 0;  
    } else {  
        System.out.println("Queue overflow");  
    }  
}
```


Queue deletion method

```
public void dequeue() {  
    if (isEmpty()) {  
        System.out.println("queue underflow");  
    }  
    else {  
        System.out.println(arr[front] + "deleted");  
    }  
    if (front == rear) {  
        front = -1;  
        rear = -1;  
    } else  
        front++;  
}
```

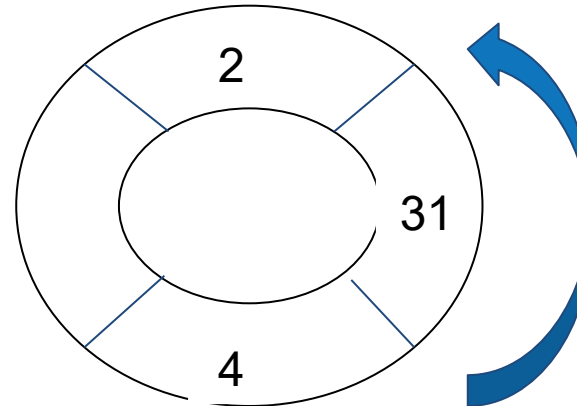
Circular Queue using array

- Here, we implement circular queue concept using array.
- We have front and rear pointers to perform queue operations in array.
- As shown below we have two types of views

How computer views



How Programmer views



Circular Queue using array example

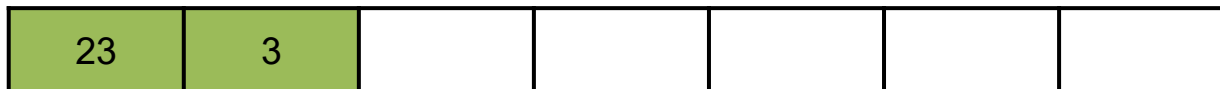
Insertion



$F = R = -1$



$F = R = 0$



$R = 0 \quad F = 1$

Initially
 $\text{Front} = \text{rear} = -1$

Circular Queue using array example

Deletion



$R = 1$

$F = 5$



$R = 2$

$F = 5$



$R = 3$

$F = 5$

Circular Queue isFull() , isEmpty() functions

```
boolean isFull() {  
    if((rear+1)%capacity==front) {  
        return true;  
    }  
    return false;  
}
```

```
boolean isEmpty() {  
    if (front == -1)  
        return true;  
    else  
        return false;  
}
```

Circular Queue insertion method

```
void enQueue(int element) {  
    if (isFull()) {  
        System.out.println("Queue is full");  
    } else {  
        if (front == -1)  
            front = 0;  
        rear = (rear + 1) % capacity;  
        arr[rear] = element;  
        System.out.println("Inserted " + arr[rear]);  
    }  
}
```

Circular Queue deletion method

```
void deQueue() {  
    if (isEmpty()) {  
        System.out.println("Queue is empty");  
    }  
    else {  
        System.out.println("element removed " + arr[front]);  
        if (front == rear) {  
            front = -1;  
            rear = -1;  
        }  
        else {  
            front = (front + 1) % capacity;  
        }  
    }  
}
```

Summary

- We have learnt what queues are and their principles and operations.
- We have learnt what circular queues are.
- We have also seen Insertion and deletion of elements in queues and circular queues using array along with algorithm and example.



Thank You