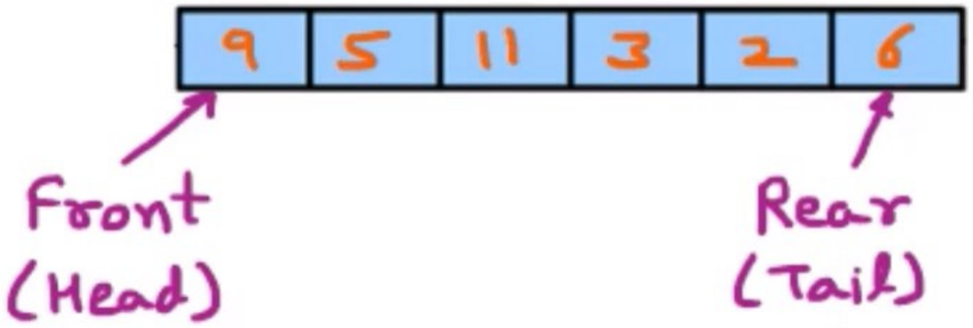
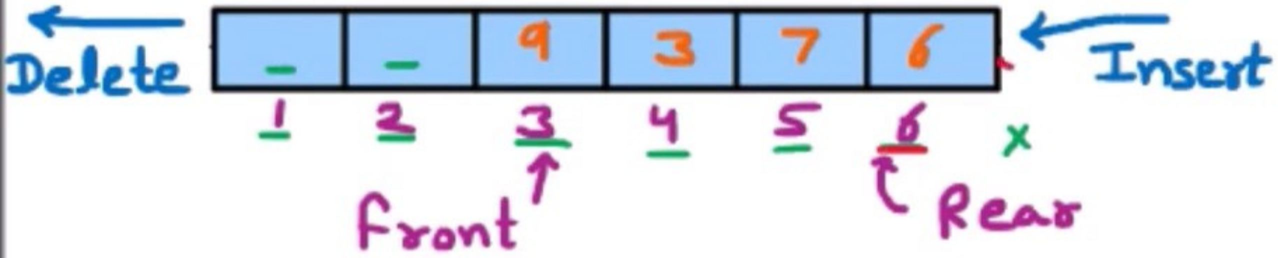
Queue

Representation

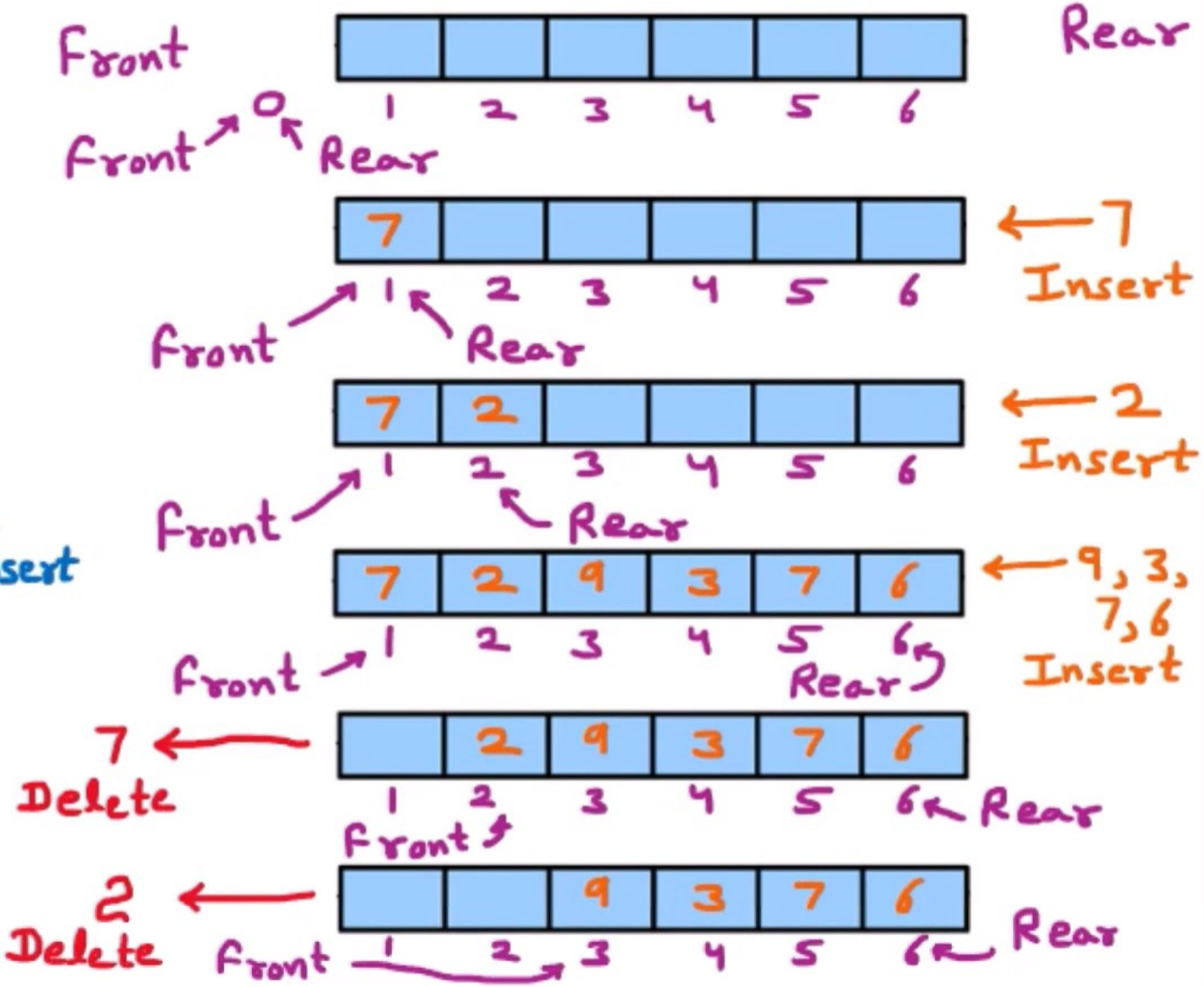


Problem



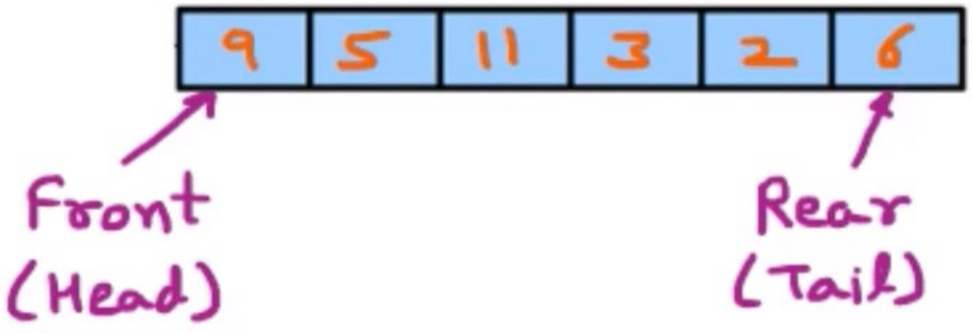
Solution: Circular Queue

Insertion & Deletion

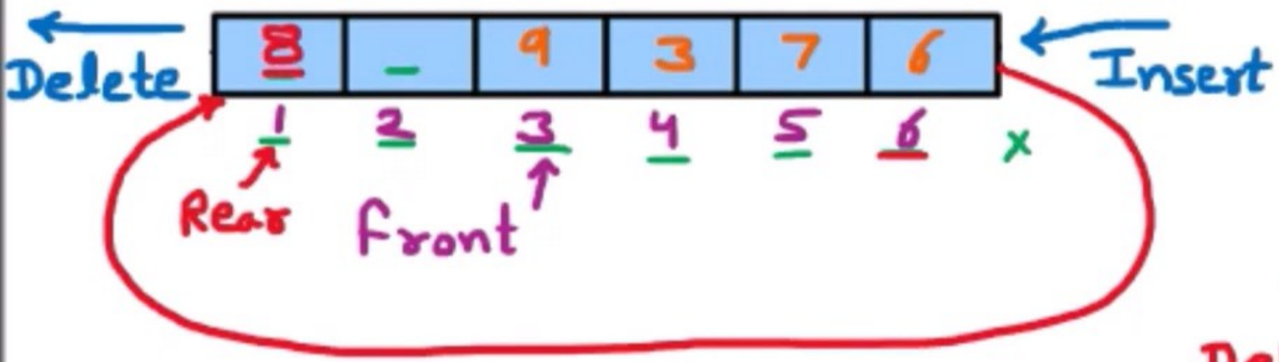


Queue

Representation

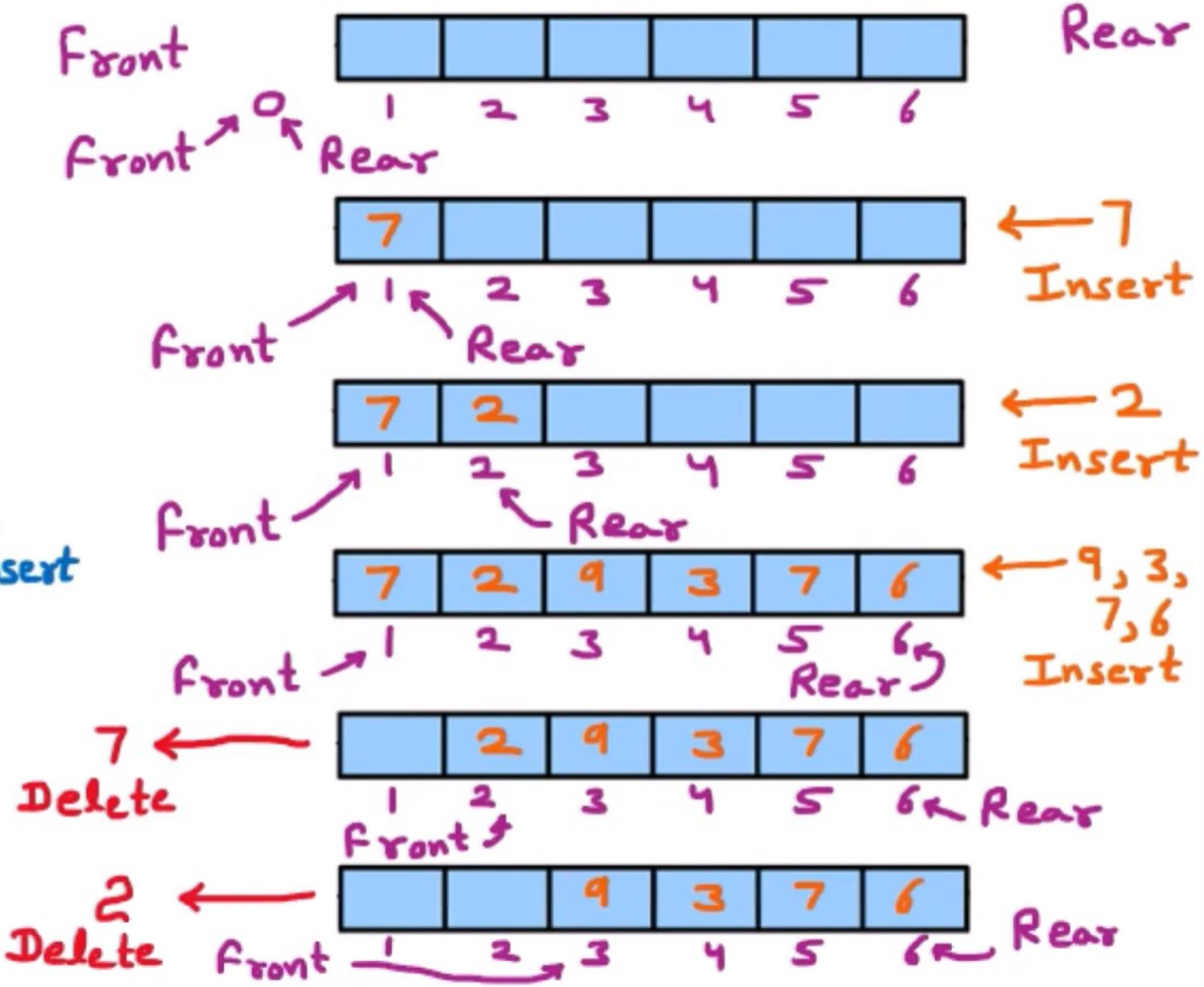


Problem



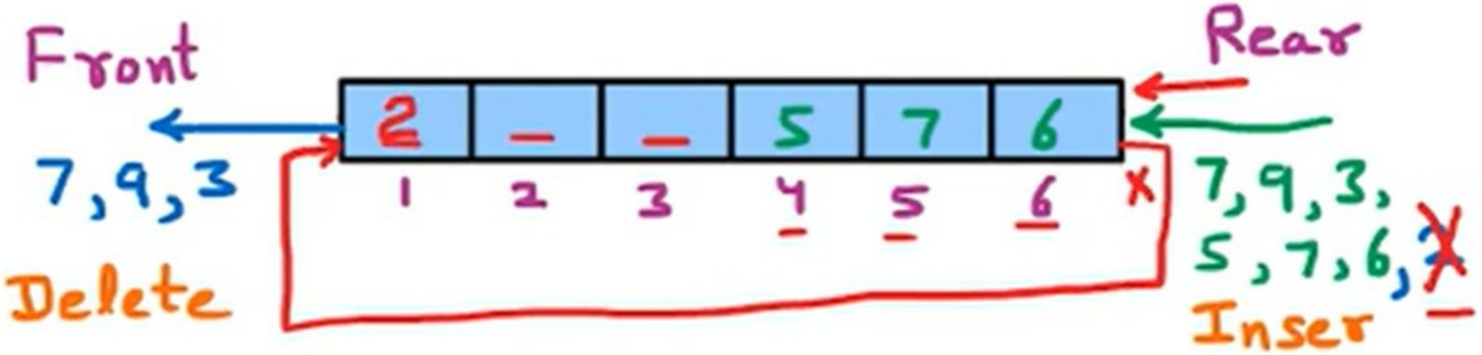
Solution: Circular Queue

Insertion & Deletion

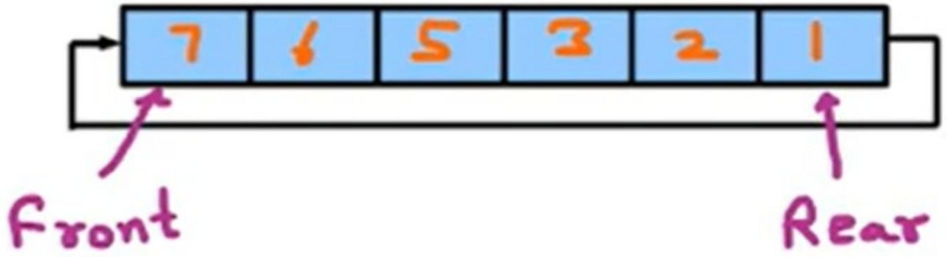


Circular Queue

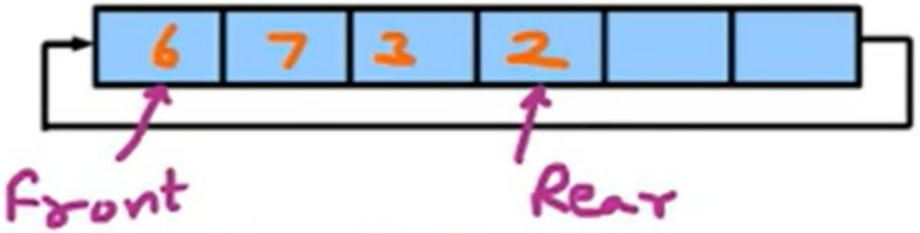
Queue



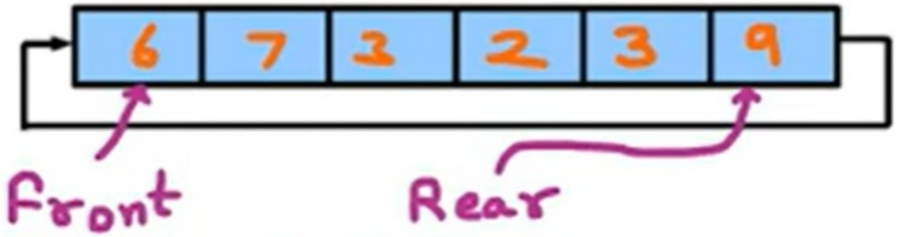
Circular Queue



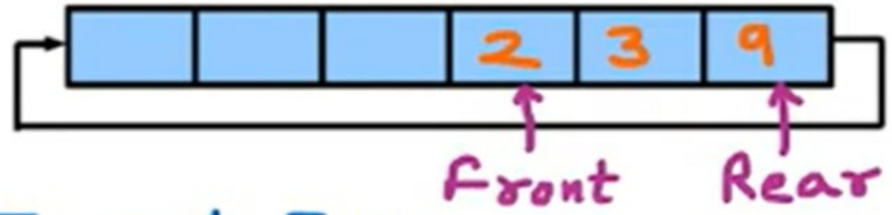
Insertion & Deletion



Insert 3,9

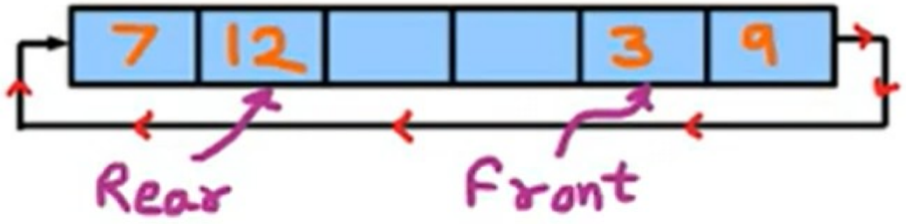


Delete 6,7,3



Insert 7,12

Delete 2



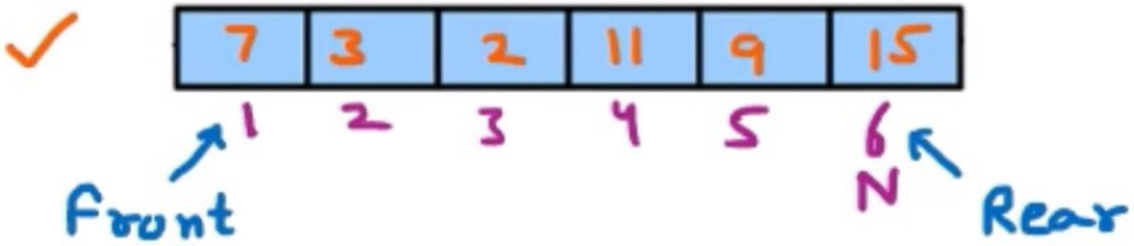
ALGORITHM: QINSERT (QUEUE, N, FRONT, REAR, ITEM)

This procedure inserts an element ITEM into a queue.

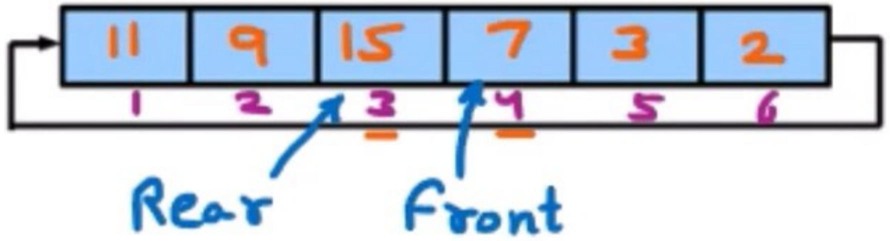
ITEM
7

```
x 1.  If FRONT = 1 and REAR = N, ~~or if FRONT = REAR + 1~~ then:
          Write: OVERFLOW, and Return.
x 2.  If FRONT := NULL, then:
          Set FRONT :=1 and REAR := 1.
      ~~Else if REAR := N, then:~~
          ~~Set REAR := 1.~~
      Else:
          Set REAR := REAR +1.
  3.  Set QUEUE [REAR] := ITEM.
  4.  Return.
```

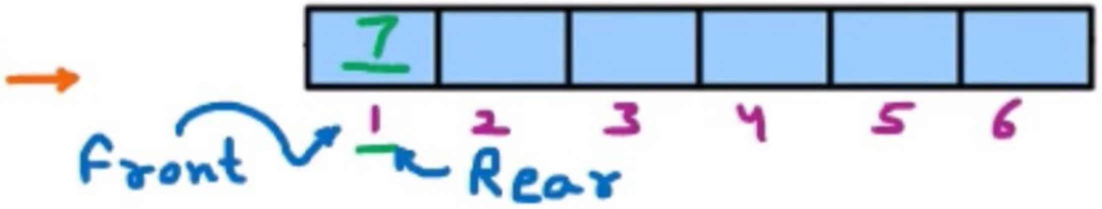
x → ✓



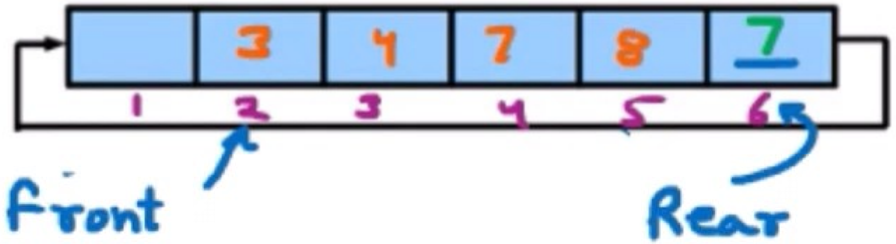
x →



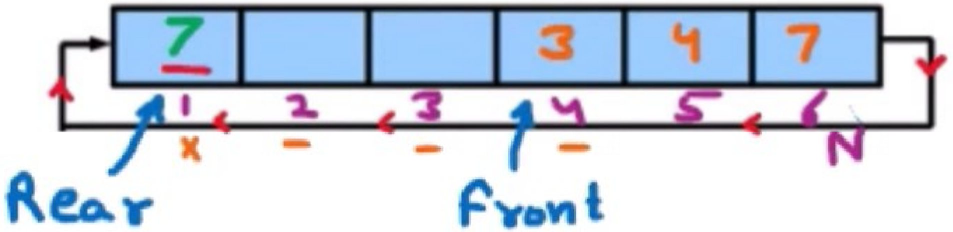
✓ →



✓ →



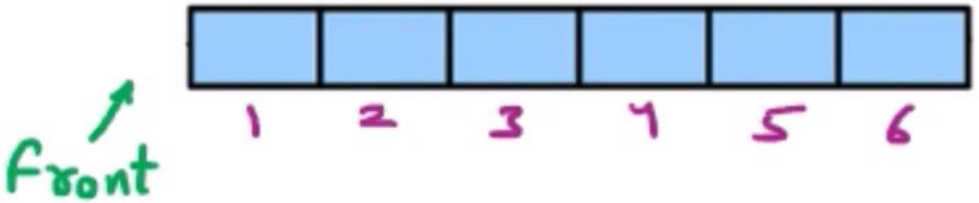
✓ →



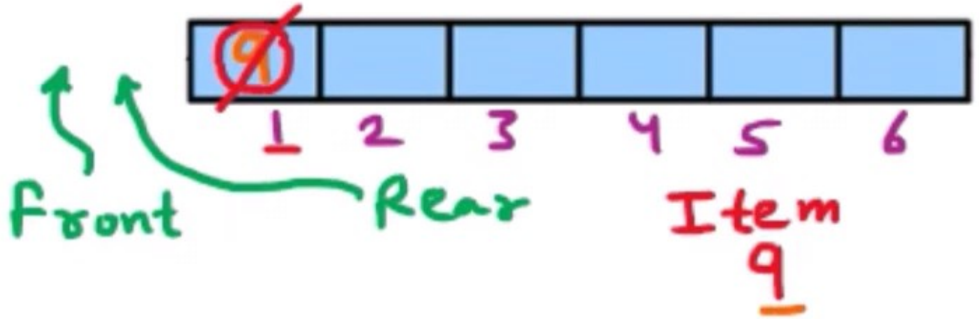
ALGORITHM: QDELETE (QUEUE, N, FRONT, REAR, ITEM)

This procedure deletes an element from a queue and assign it to the variable ITEM.

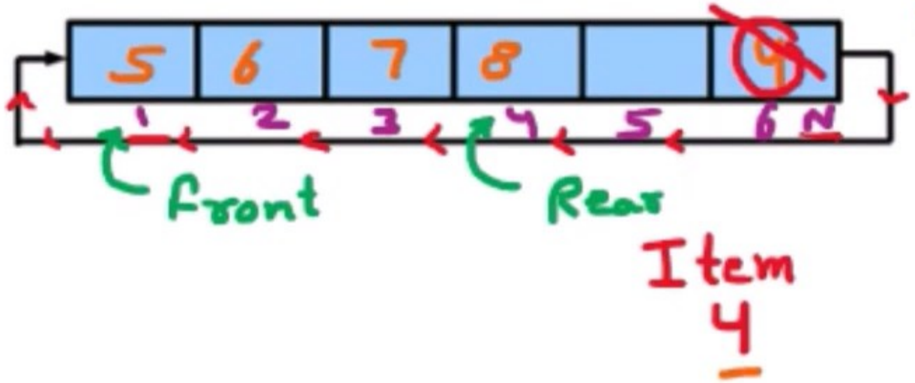
```
1.  If FRONT := NULL, then:                          X
        Write: UNDERFLOW, and Return.
2.  Set ITEM := QUEUE [FRONT]. ✓
3.  If FRONT := REAR, then:
        Set FRONT := NULL and REAR := NULL.          X
    Else if FRONT := N, then:
        Set FRONT := 1.
    Else:
        Set FRONT := FRONT +1.                       X
3.  Return.
```



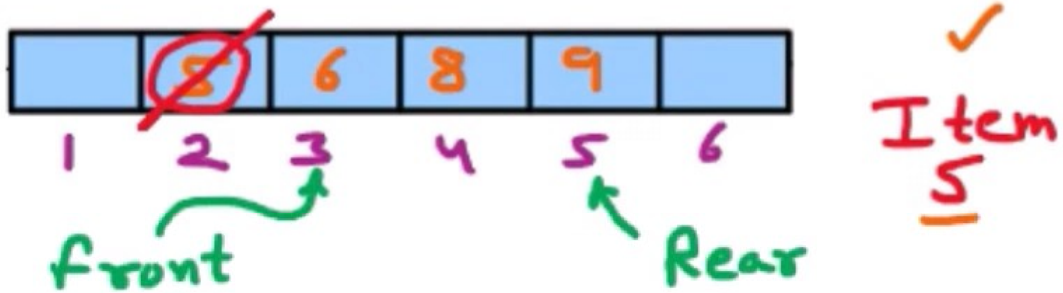
✓



✓



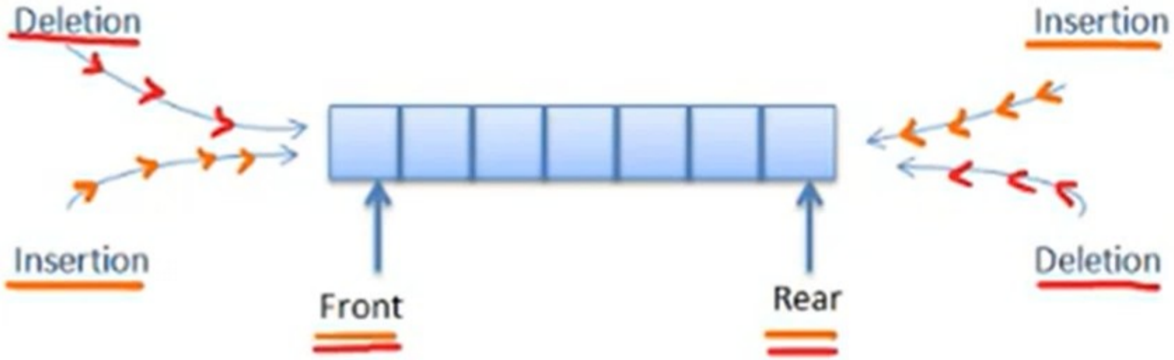
✓



✓

Double Ended Queue (DEQUE)

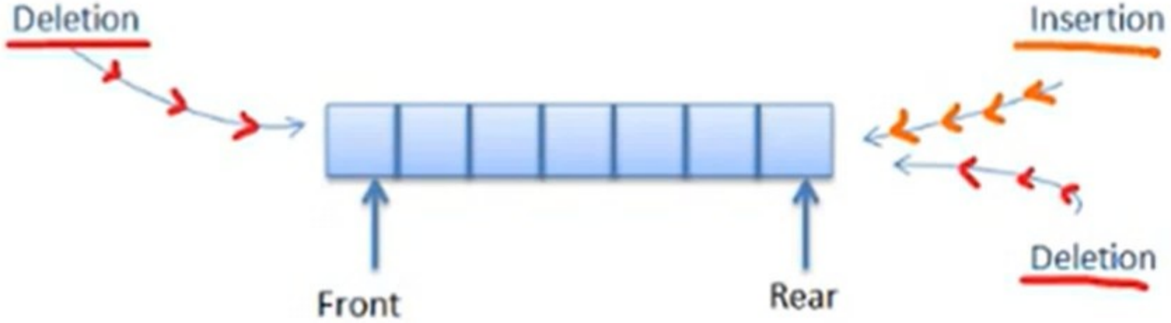
DEQUE is a Linked List in which elements can be Inserted or Deleted at either end but not the middle



Two Variants of DEQUE

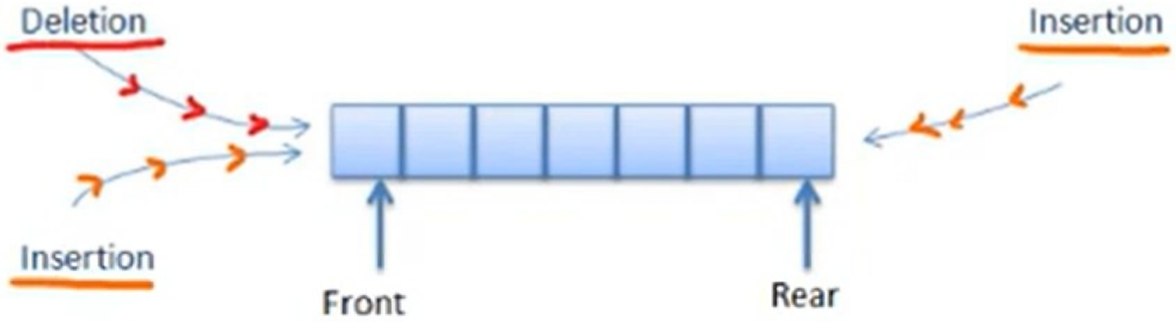
Input Restricted Queue

- Allow Insersion at one end but allow Deletion at both ends

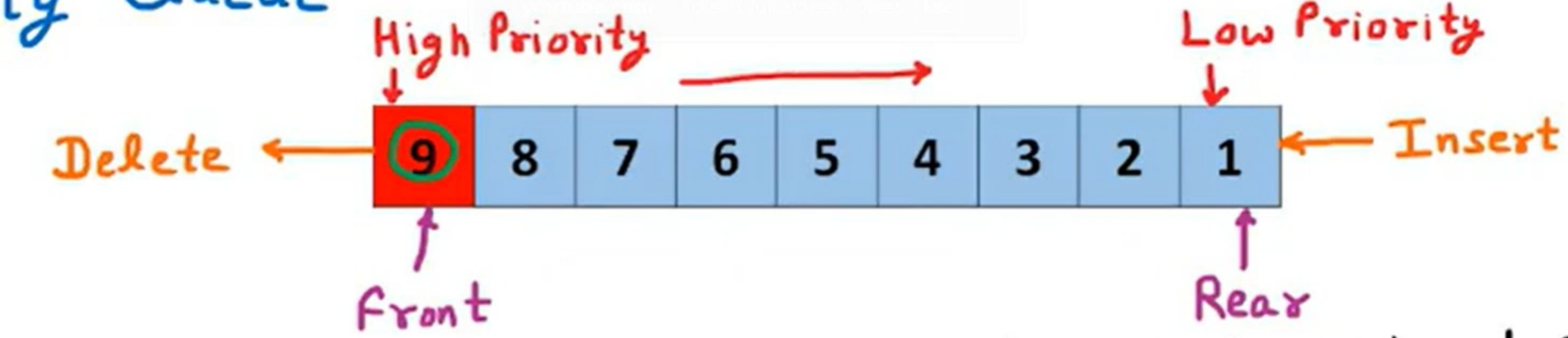


Output Restricted Queue

- Allow Deletion at one end but allow Insertion at both ends



Priority Queue



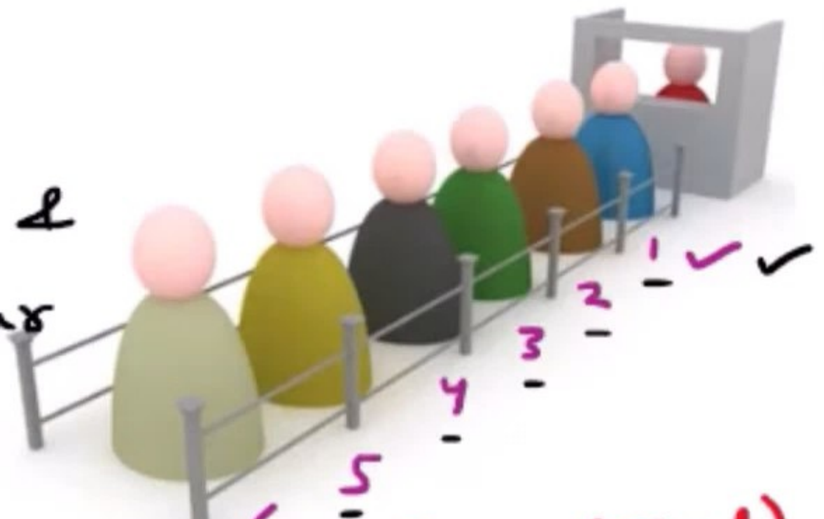
- Collection of elements in which each element is assigned priority
- According to priority elements are deleted & processed as per following rules
  → Highest Priority element is processed before Lower Priority Element
  → Two elements with same priority are processed according to order in which they are added to queue

Example:

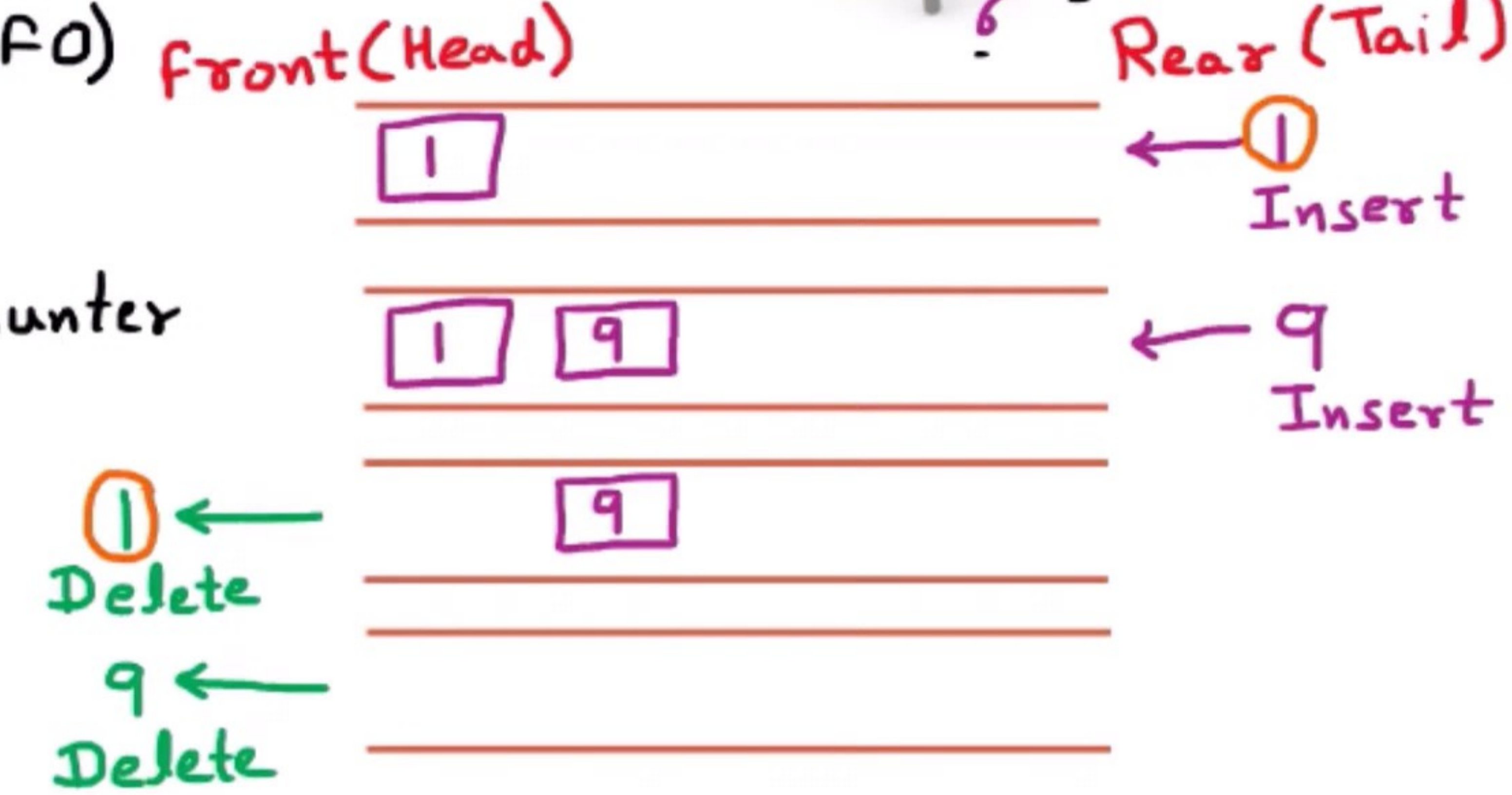
Time Sharing System

Queue

- Queue is linear list of elements where Deletion takes place at one end called front & Insertion takes place at another end called Rear



- First In first Out (FIFO)



Examples

- Customers at ticket counter
- Automobiles waiting in line
- Programs waiting to be executed