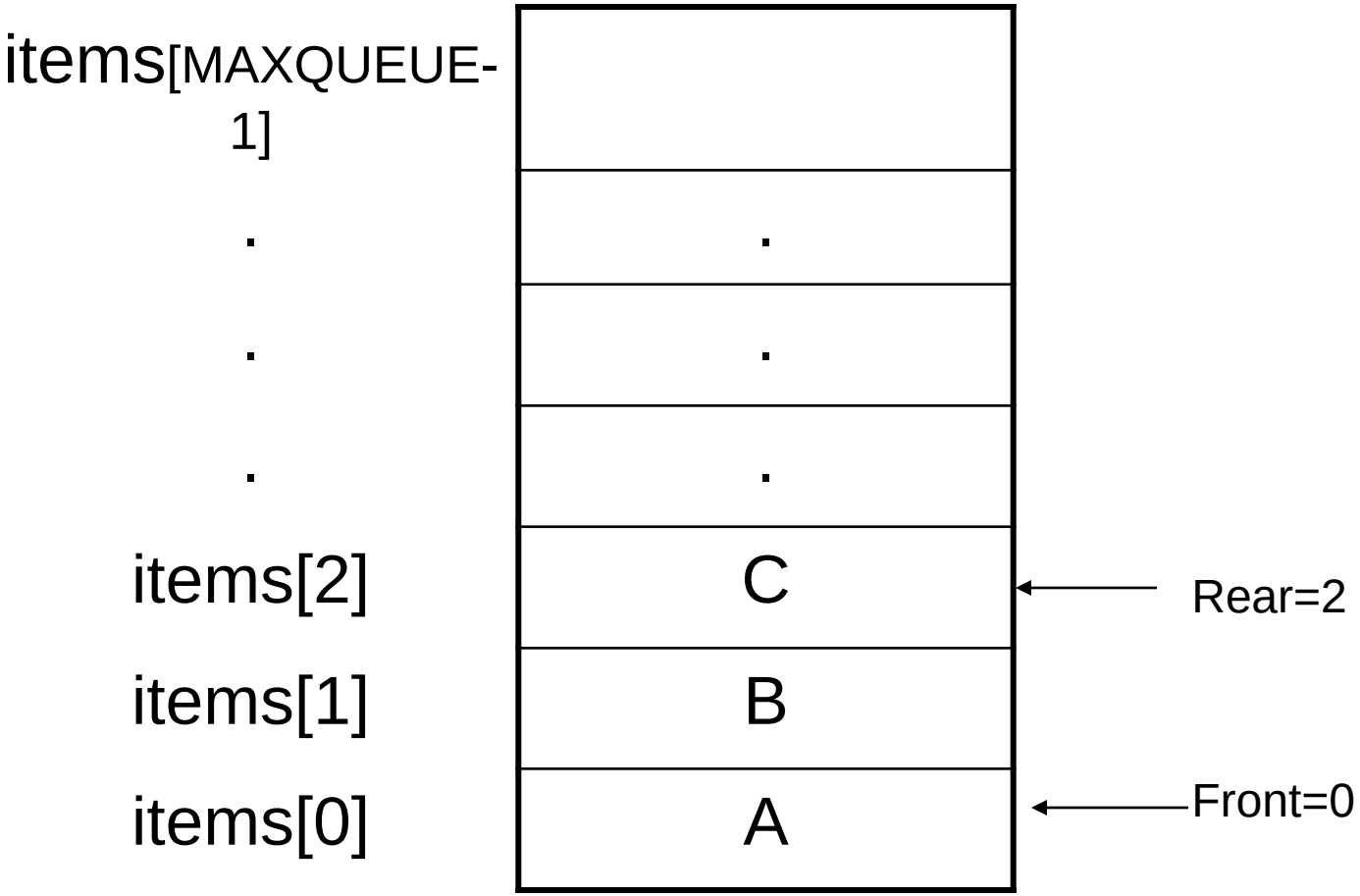
Queues

- Queses:
  - Array and list representation,
  - Operations (traversal, insertion and deletion)
- Priority queues and Deques:
  - Array and List representations

Definition of Queue

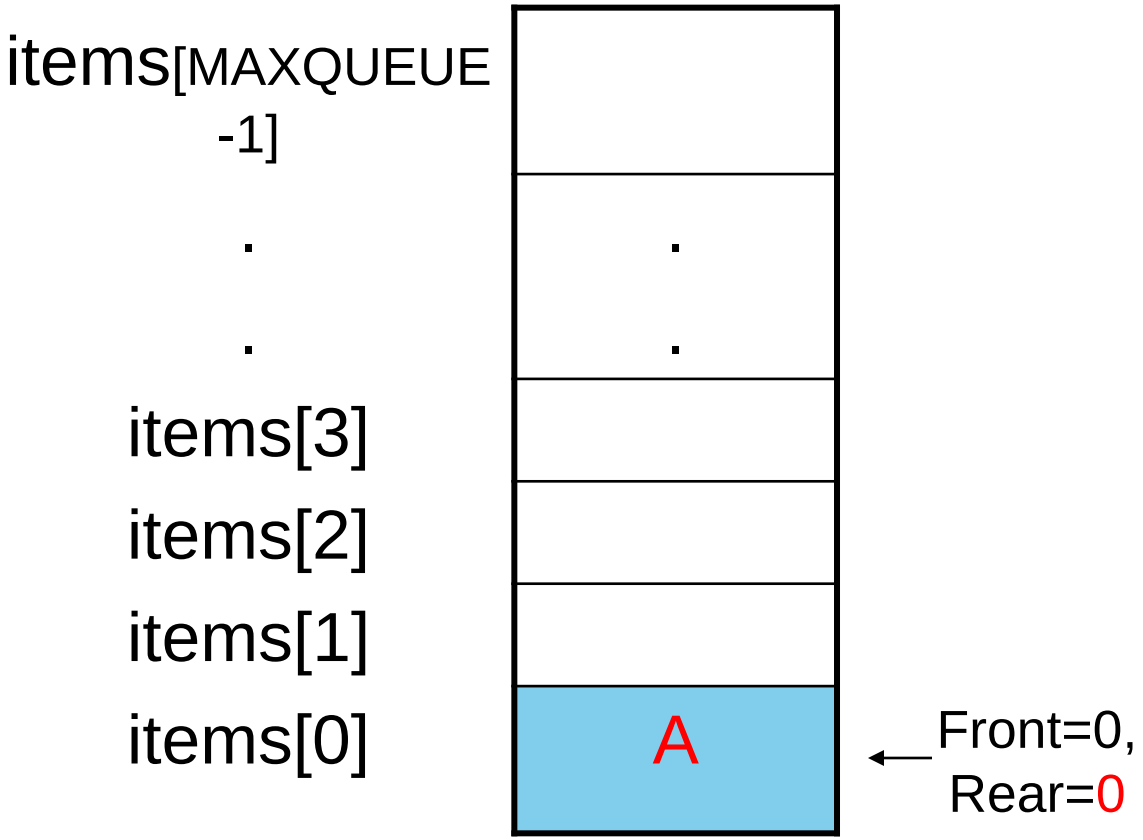
- A **Queue** is an ordered collection of items from which items may be deleted at one end (called the ***front*** of the queue) and into which items may be inserted at the other end (the ***rear*** of the queue).
- The first element inserted into the queue is the first element to be removed. For this reason a queue is sometimes called a **FIFO** (first-in first-out) list as opposed to the stack, which is a **LIFO** (last-in first-out).

Queue



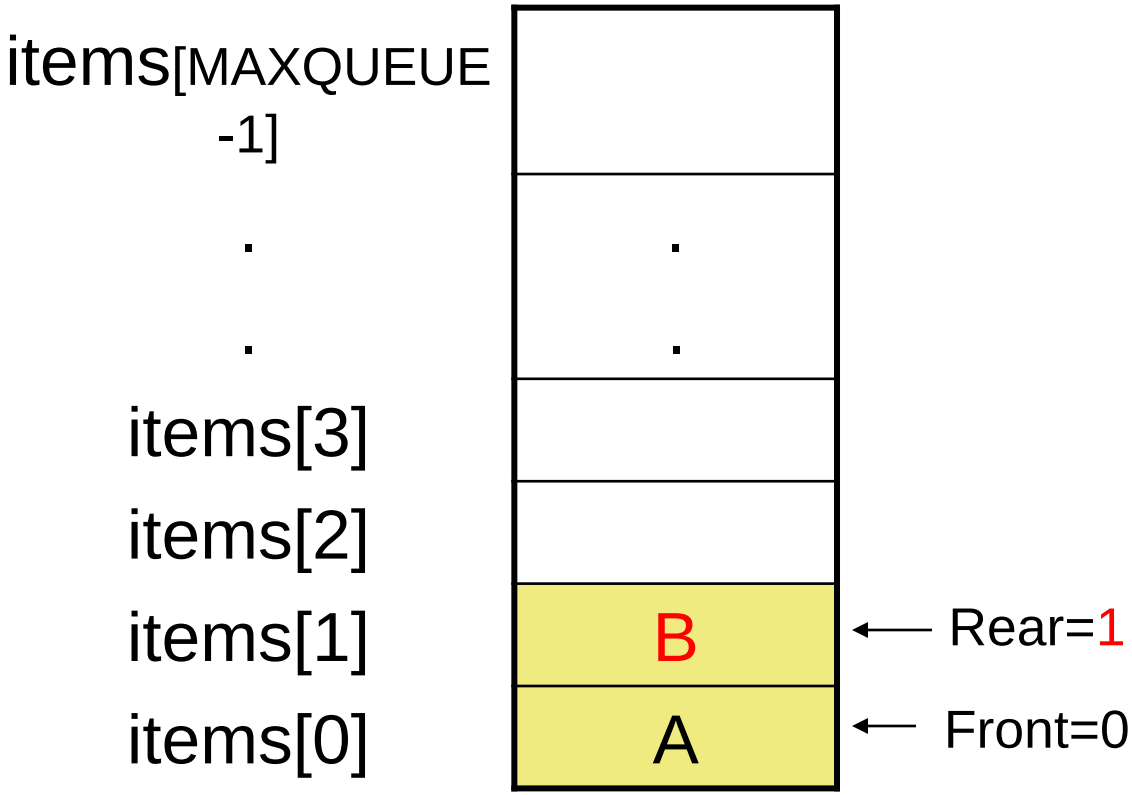
Insert an item A

- **A new item (*A*) is *inserted* at the *Rear* of the queue**



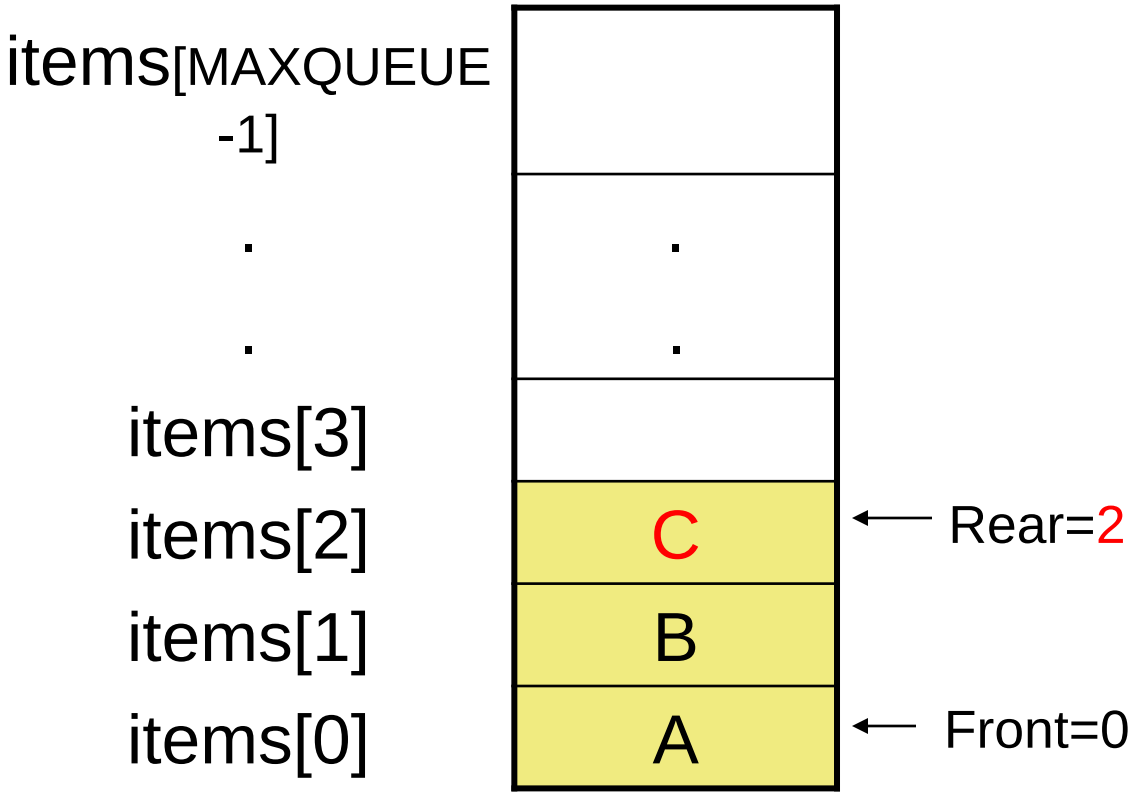
Insert an item B

- **A new item (*B*) is *inserted* at the *Rear* of the queue**



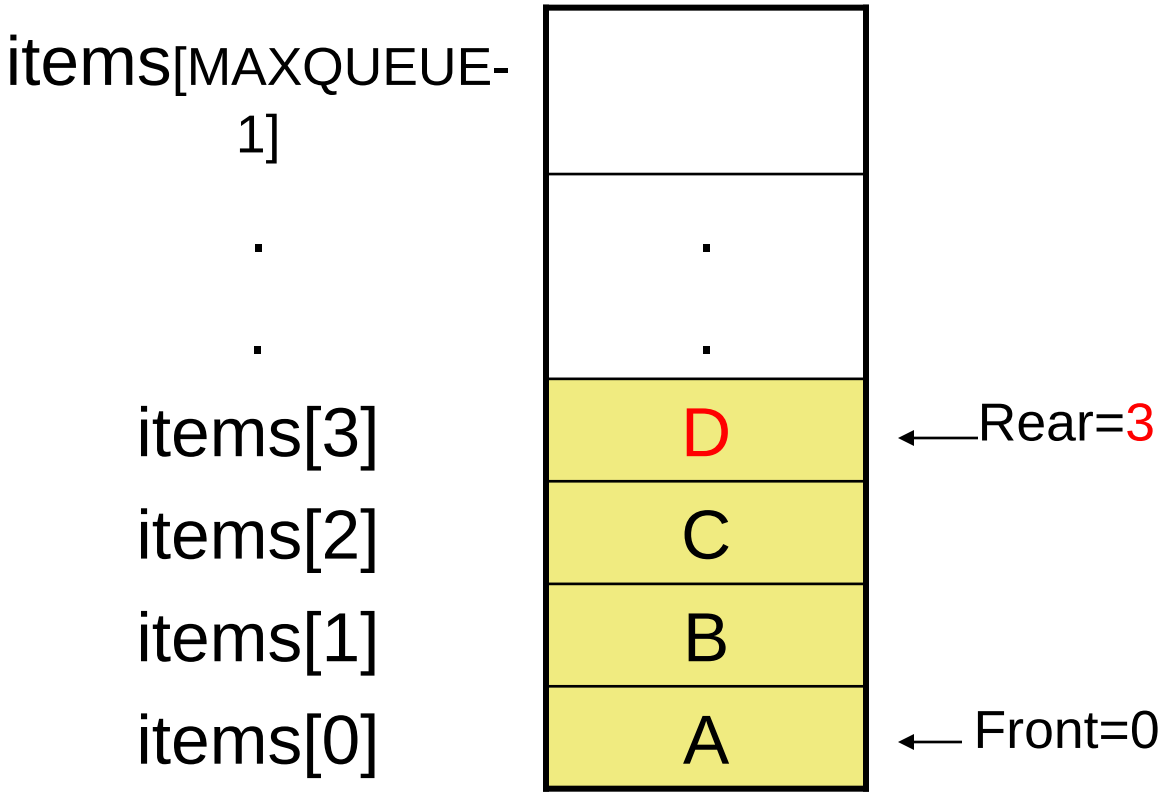
Insert an item C

- **A new item (*C*) is *inserted* at the *Rear* of the queue**



Insert an item D

- **A new item (*D*) is *inserted* at the *Rear* of the queue**



Insert Operation(Array)

```
Step 1: IF REAR = MAX-1
            Write OVERFLOW
            Goto step 4
        [END OF IF]
Step 2: IF FRONT = -1 and REAR = -1
            SET FRONT = REAR = 0
        ELSE
            SET REAR = REAR + 1
        [END OF IF]
Step 3: SET QUEUE[REAR] = NUM
Step 4: EXIT
```

Circular Queue Insertion using Array

1.F=1 and R=N or if F=R+1 then Write: OverFlow and return

2.If F=NULL then set F=1and R=1

Elseif R=N then set R=1

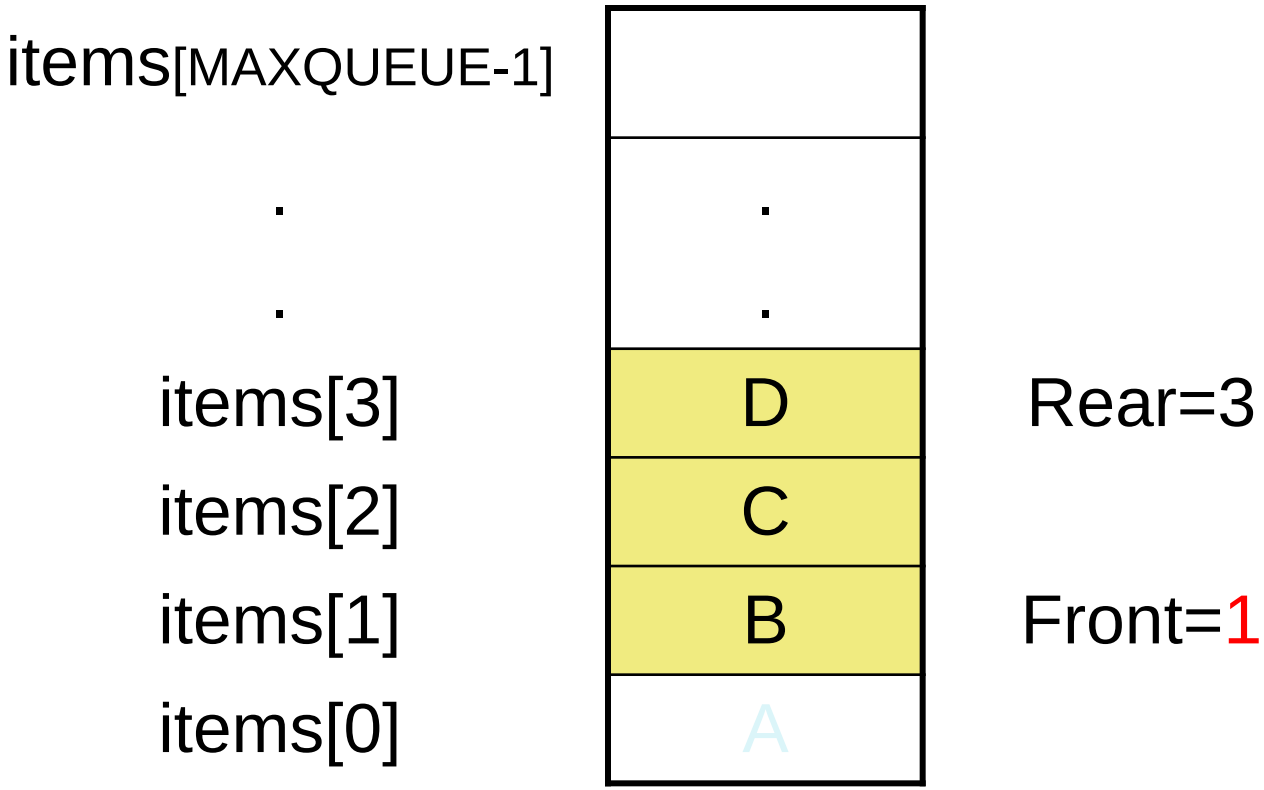
Else Set R=R+1

3.Set Q[R]=ITEM

4.Return

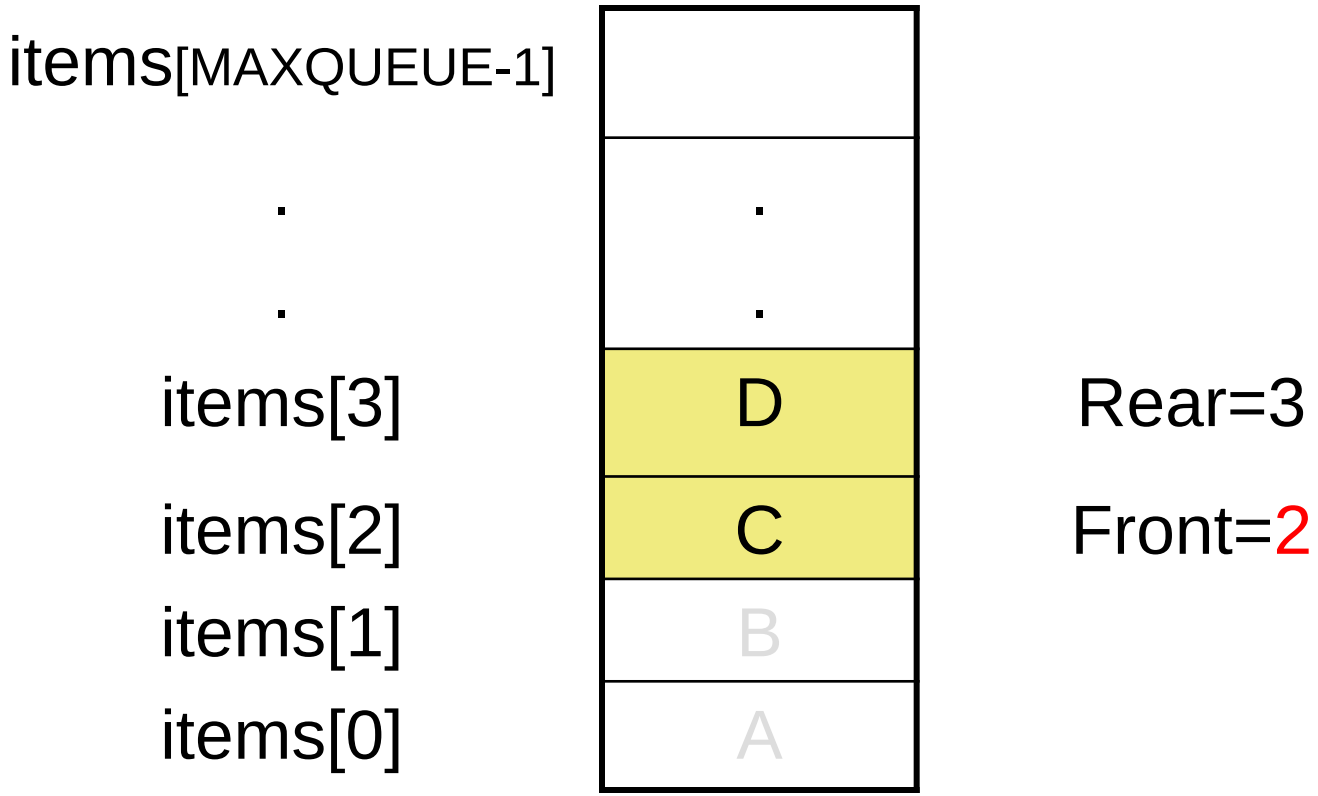
Delete A

- **An item (*A*) is deleted from the *Front* of the queue**



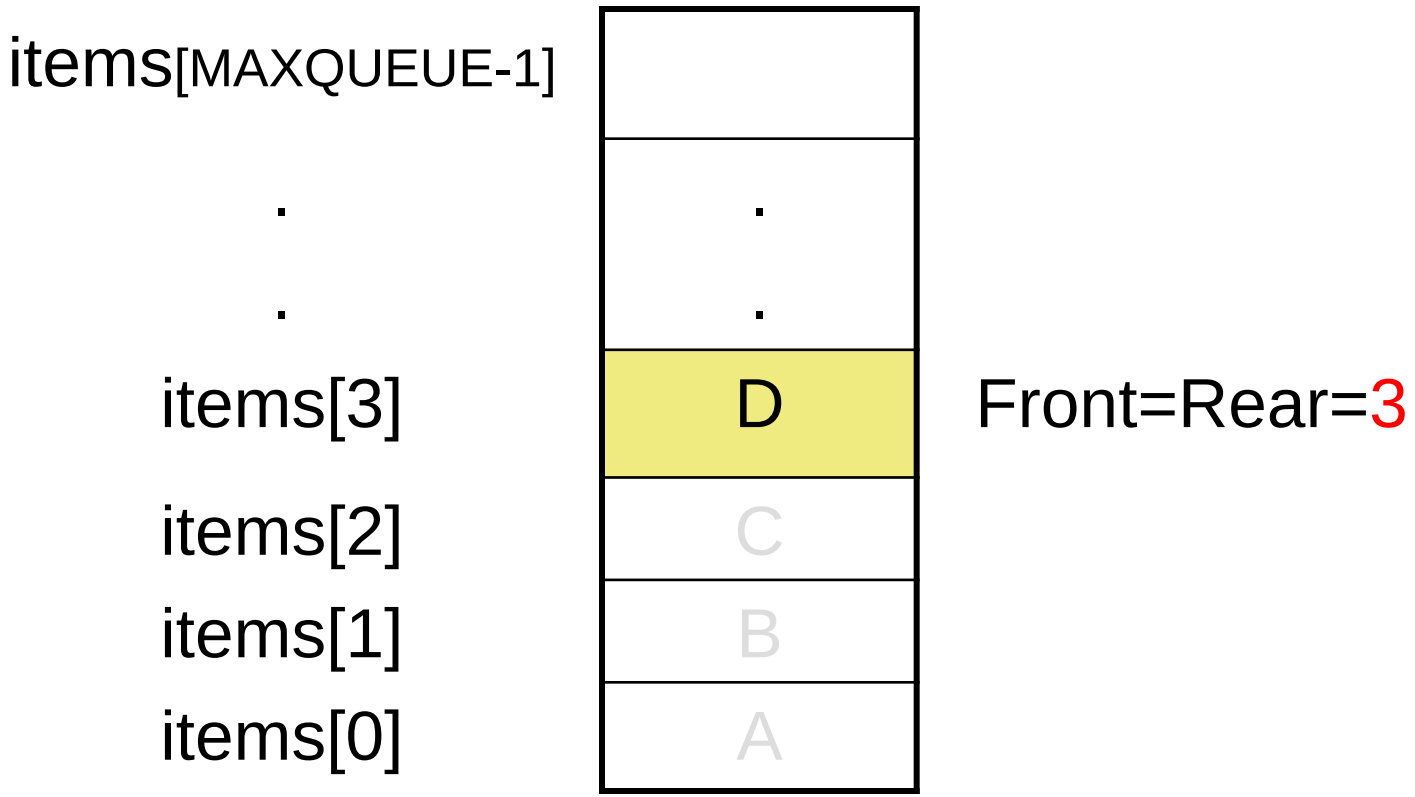
Delete B

- **An item (*B*) is deleted from the *Front* of the queue.**



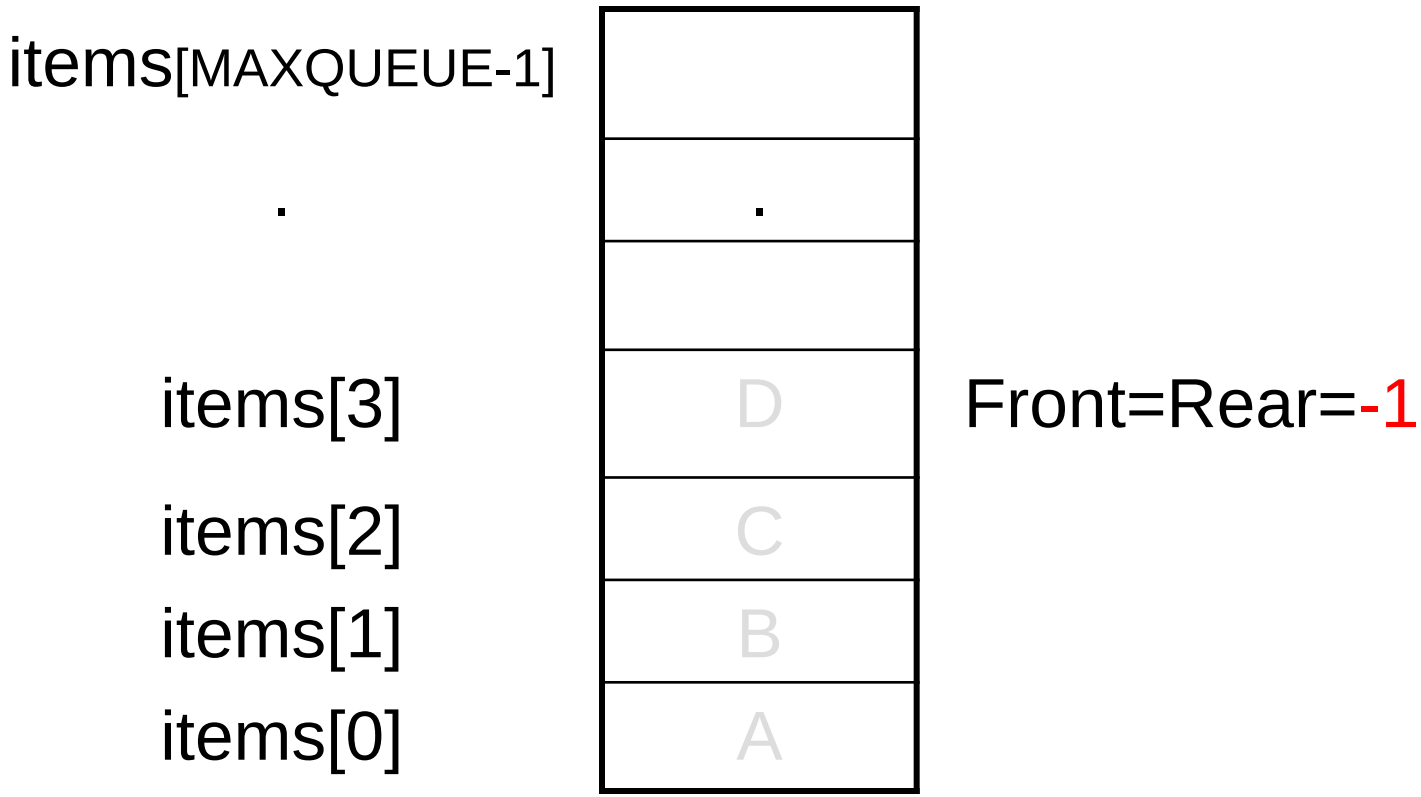
Delete C

- **An item (*C*) is deleted from the *Front* of the queue**



Delete D

- **An item (*A*) is deleted from the *Front* of the queue.**



Delete Operation(Array)

```
Step 1: IF FRONT = -1 OR FRONT > REAR
            Write UNDERFLOW
        ELSE
            SET VAL = QUEUE[FRONT]
            SET FRONT = FRONT + 1
        [END OF IF]
Step 2: EXIT
```

Circular Queue Deletion using Array

1.F=NULL then Write: UnderFlow and return

2. Set ITEM=Q[F]

3. If F=R then set F=NULL and R=NULL

Elseif F=N then set F=1

Else Set F=F+1

4.Return

Insertion in Queue using LL

1. If AVAIL=NULL then Write OVERFLOW and Exit
2. Set NEW=AVAIL & AVAIL=LINK[AVAIL]
3. Set INFO[NEW]=ITEM & LINK[NEW] =NULL
4. If (FRONT=NULL then FRONT=REAR=NEW

Else Set LINK[REAR]=NEW & REAR=NEW

4.Return

Deletion in Queue using LL

1. If FRONT=NULL then Write UNDERFLOW and Exit
2. Set TEMP=FRONT
3. ITEM=INFO[FRONT]
4. FRONT=LINK[TEMP]
5. LINK[TEMP]=AVAIL & AVAIL=TEMP
6. EXIT

Priority Queue

- More specialized data structure.
- Similar to Queue, having front and rear.
- Items are removed from the front.
- Items are ordered by key value so that the item with the lowest key (or highest) is always at the front.
- Items are inserted in proper position to maintain the order.

Deques

- It is a double-ended queue.
- Items can be inserted and deleted from either ends.
- More versatile data structure than stack or queue.
- E.g. policy-based application (e.g. low priority go to the end, high go to the front)
- Two variations: Input restricted(allow insertion at only one end but allow deletions at both ends) and output restricted(restricted(allow deletion at only one end but allow insertions at both ends)

Thank You