QUEUE

Contents

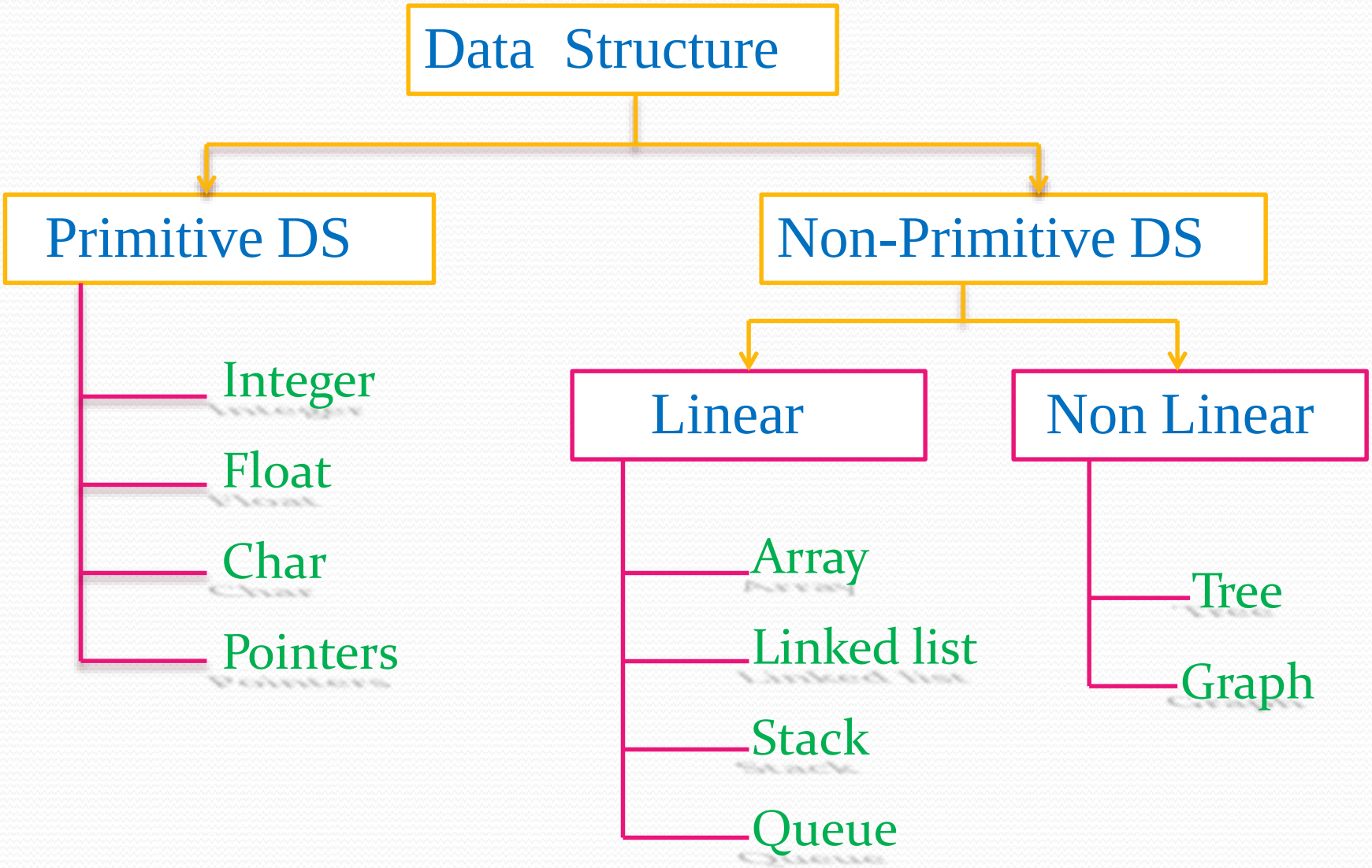
- Introduction
- Operations on queue
- Array representation of queues
- Linked representation of queues
- Types of queues
  - Circular queues
  - Deques
  - Priority queues
- Application of queues
- References

DATA STRUCTURE

- A data structure is a particular way of organizing data in a computer so that it can be used efficiently.

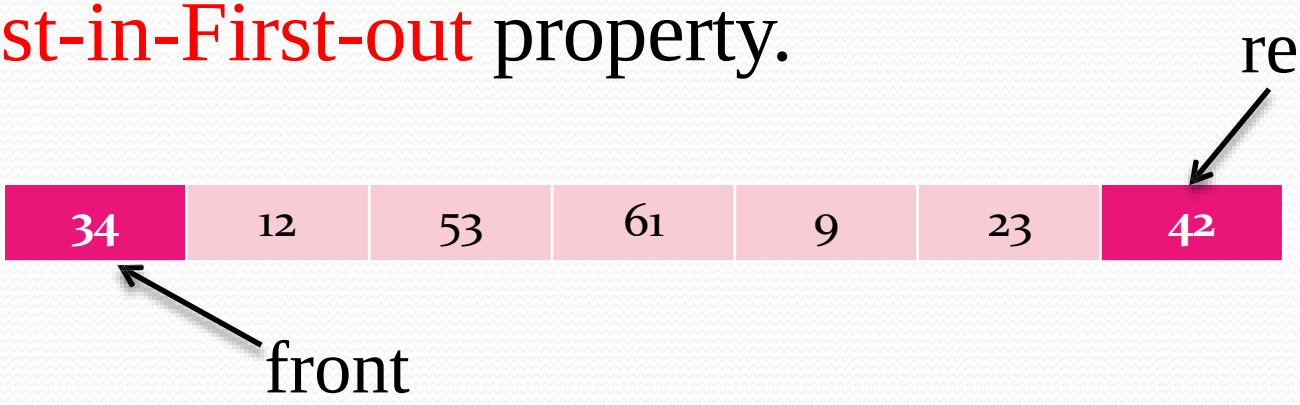
- Different kind of data structure suits for the different kind of applications.

Types of data structure



QUEUE

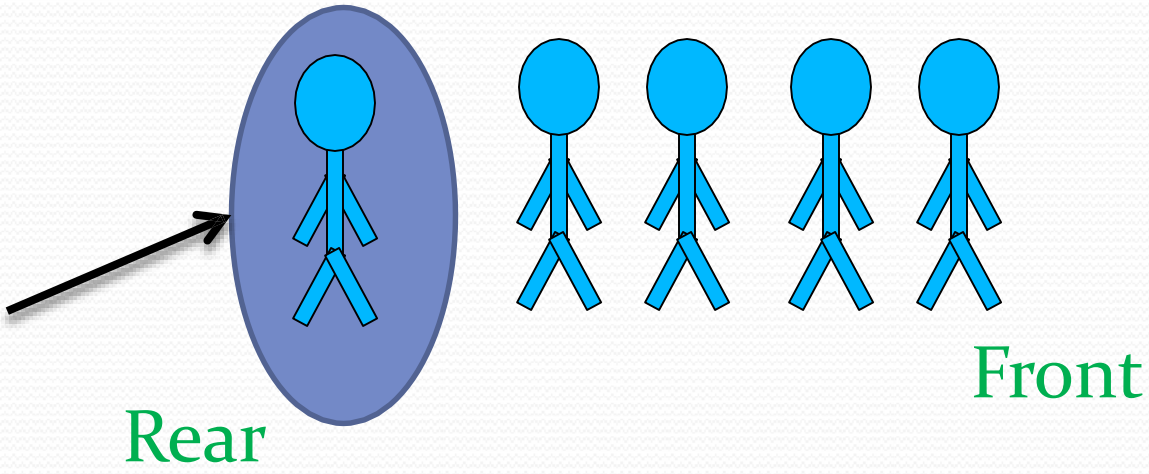
- Queue is a linear data structure.
- It is used for temporary storage of data values.
- A new element is added at one end called rear end.
- The existing elements deleted from the other end called front end.
- First-in-First-out property.



Operations on Queue

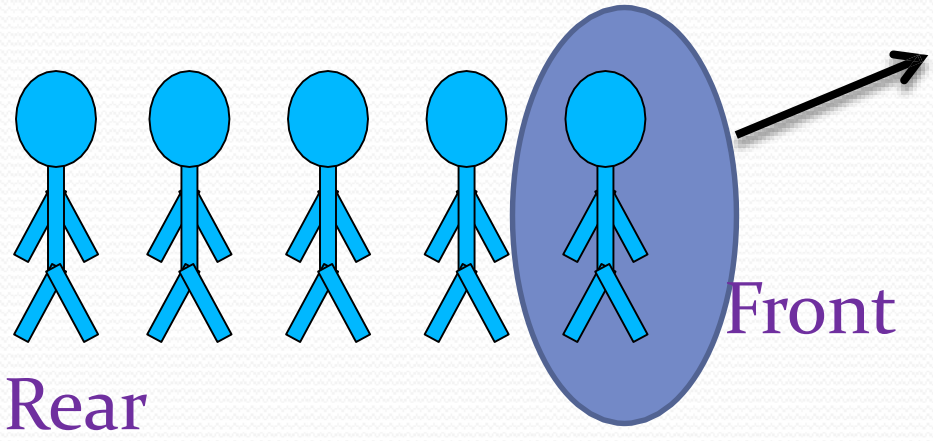
1.Insertion :

Placing an item in a queue is called “insertion or enqueue”, which is done at the end of the queue called “rear”.

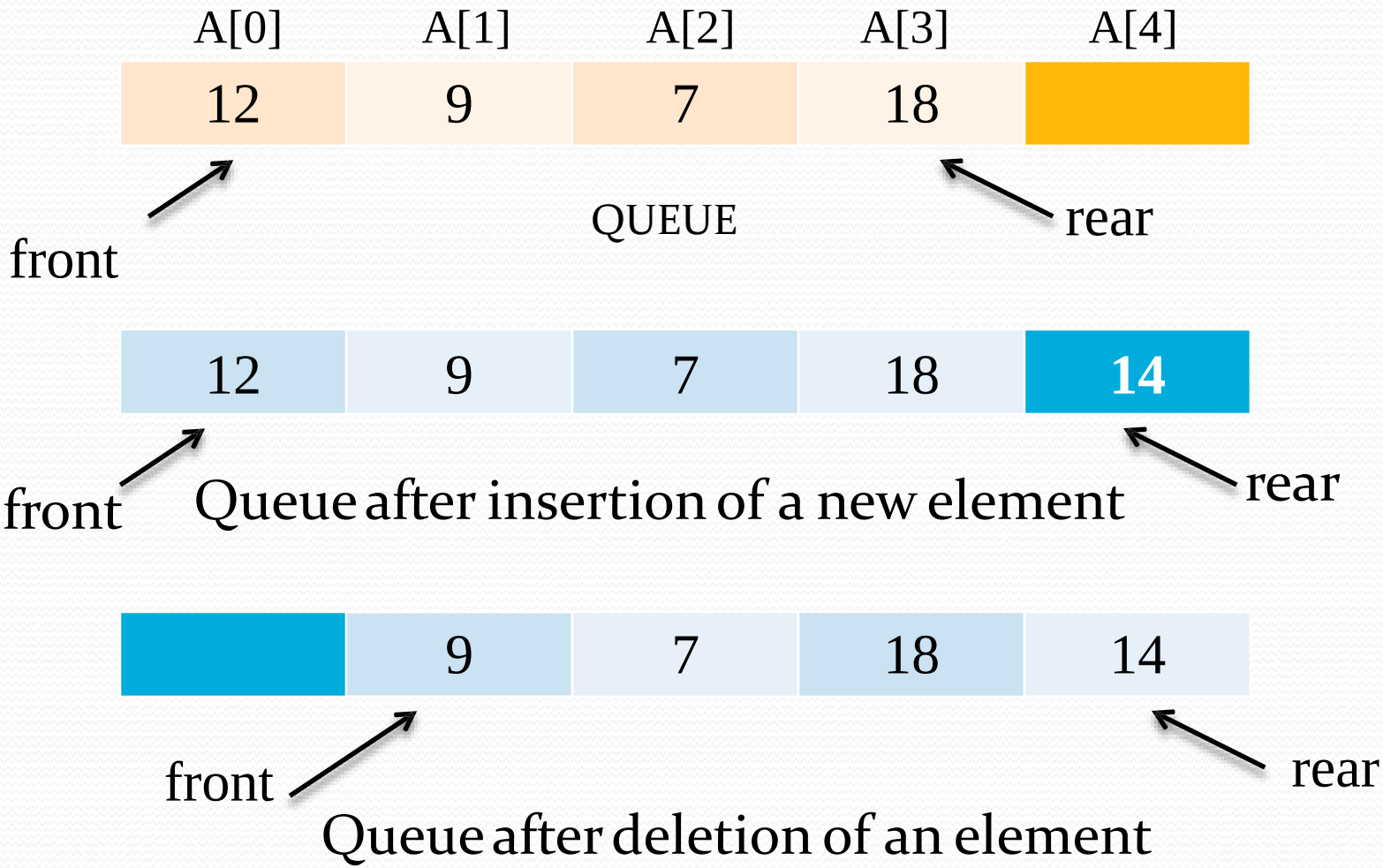


2.Deletion :

Removing an item from a queue is called "deletion or dequeue" , which is done at the other end of the queue called "front".



Array Representation of Queues



Algorithm to insert an element in queue

```
STEP-1  IF REAR= MAX-1
                write OVERFLOW
                 go to step 4
            [end of if]
STEP-2   if  REAR = -1
                 set  FRONT=REAR= 0
            else
                 set  REAR = REAR+1
STEP-3     set  QUEUE [ REAR ] = NUM
STEP-4    EXIT
```

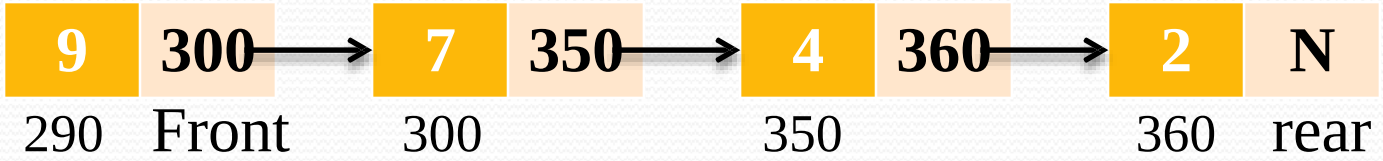
INITIALLY
REAR = -1
FRONT =-1

Algorithm to delete an element from queue

STEP-1 If FRONT = -1 or FRONT > REAR
write UNDERFLOW
Else
set VAL = QUEUE [FRONT]
set FRONT = FRONT + 1
[end of if]

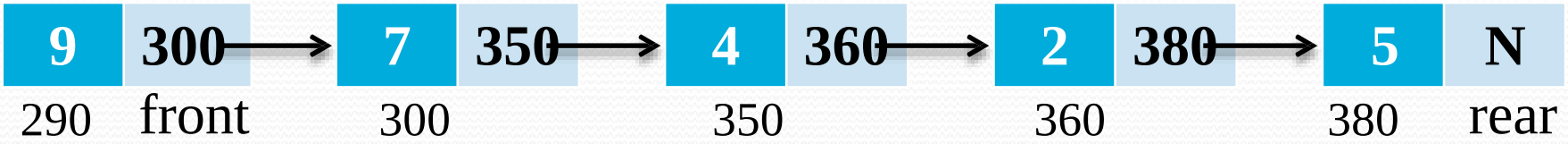
STEP-2 EXIT

Linked Representation of Queues

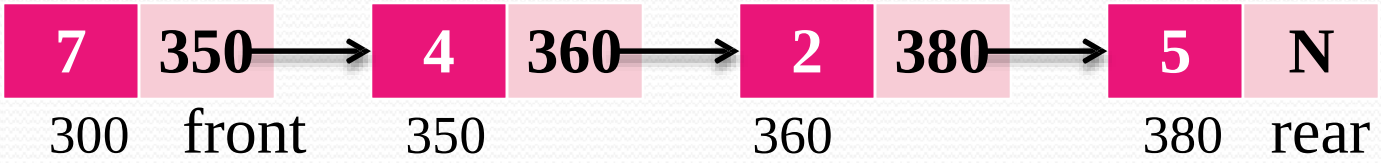


Front = 290
Rear = 360

Linked queue



Linked queue after inserting a node



Linked queue after deleting a node

Algorithm to insert an element in queue using linked list

STEP-1 Allocate memory for the new node & name it as TEMP.

STEP-2 set TEMP⟶ data = NUM

set TEMP⟶ link = NULL

STEP-3 If FRONT = NULL

FRONT = REAR = TEMP

Else

REAR ⟶ link = TEMP

REAR = TEMP

[end of if]

STEP-4 EXIT

INITIALLY
FRONT=NULL
REAR=NULL

Algorithm to delete an element from queue

STEP-1 If FRONT = NULL
write underflow
go to step 3.
[end of if]
STEP-2 set TEMP = FRONT
FRONT = FRONT⟶ link
if FRONT = NULL
REAR = NULL
STEP-3 EXIT

Types of Queues

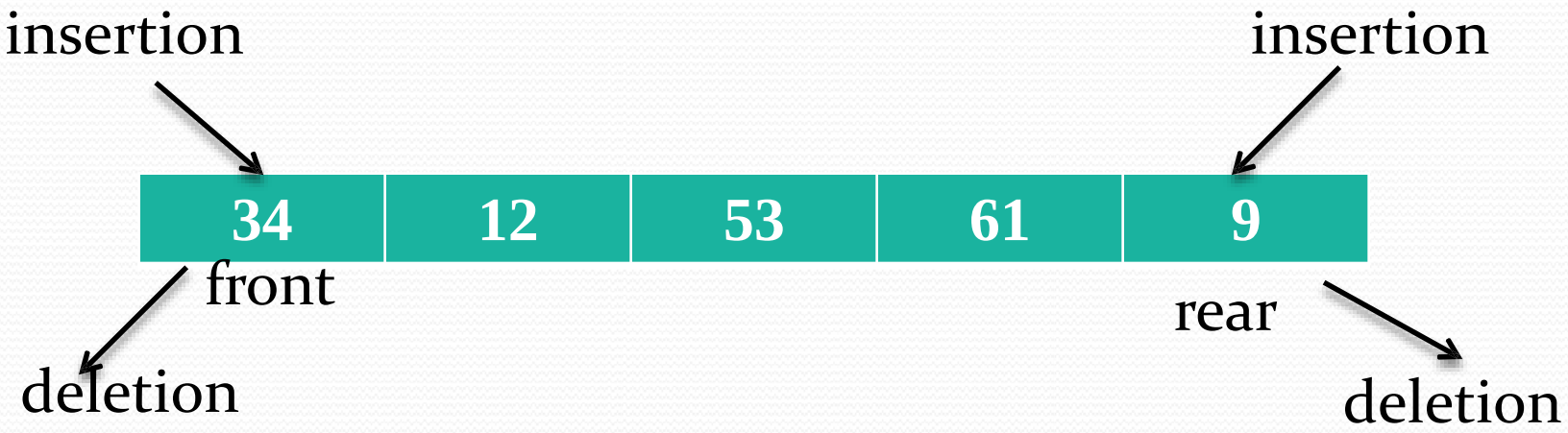
1. Deque
2. Circular Queue
3. Priority Queue

DEQUES

1.Deque stands for *double ended queue*.

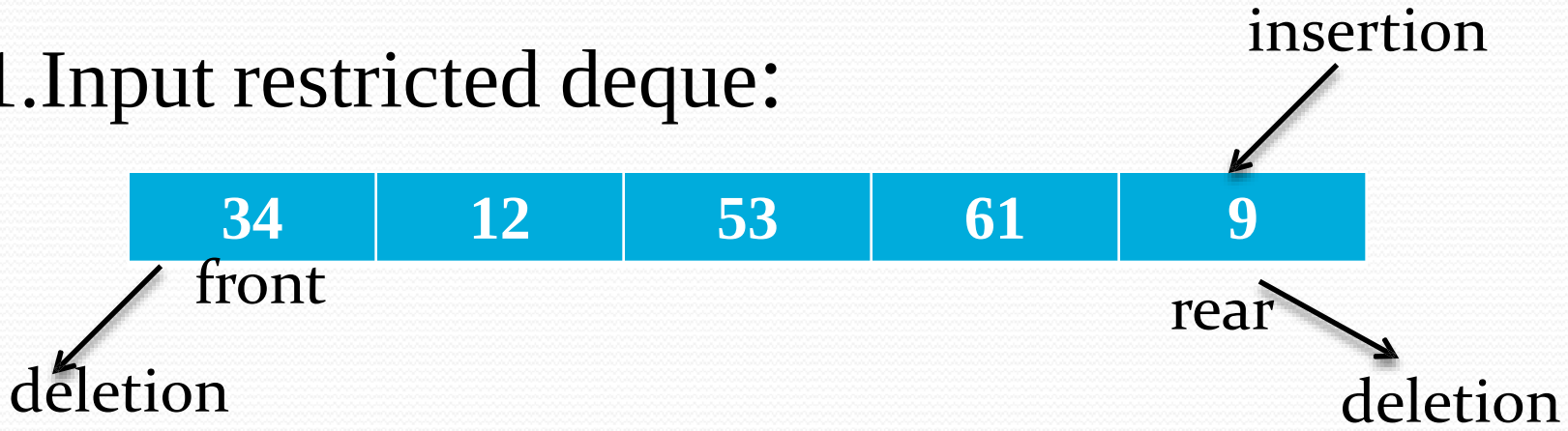
2.Elements can be inserted or deleted at either end.

3. Also known as *head-tail linked list*.

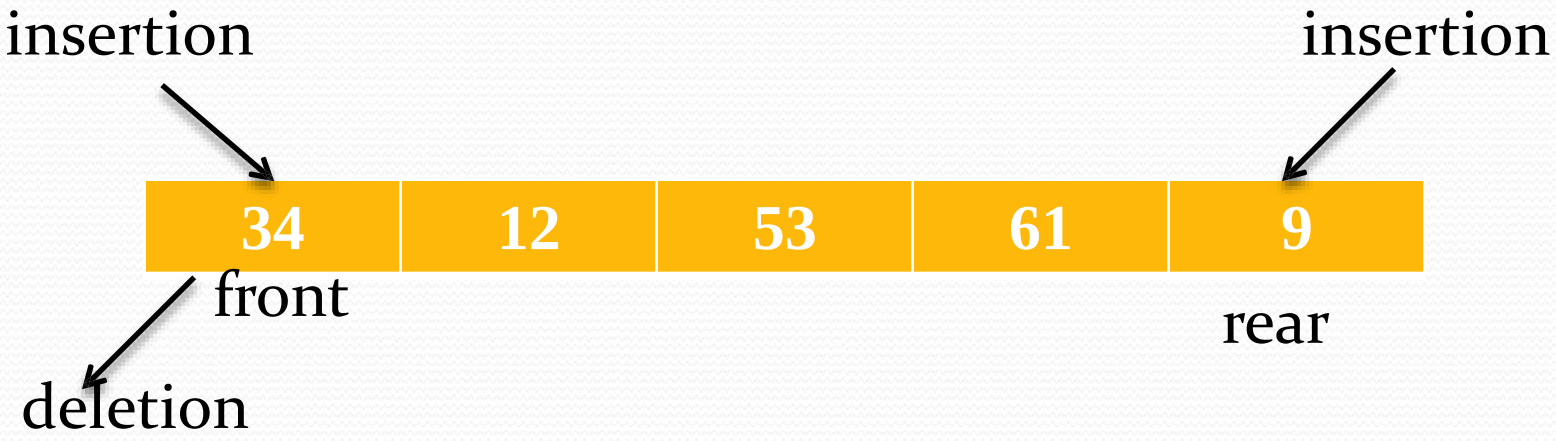


Types Of Deque

1.Input restricted deque:



2. Output restricted deque:



CIRCULAR QUEUES

- Circular queue are used to remove the drawback of simple queue.

- Both the front and the rear pointers wrap around to the beginning of the array.

- It is also called as "***Ring buffer***".

Algorithm to insert an element in queue

```
STEP-1  If FRONT = (REAR+1)%MAX
            write  OVERFLOW
            go to step 4
        [end of if]
STEP-2   If  FRONT = -1
             REAR = FRONT = 0
         Else REAR = (REAR +1)%MAX
             [ end of if ]
STEP-3    CQ[REAR] = NUM
STEP-4    EXIT
```

INITIALLY
FRONT= -1
REAR= 0

Algorithm to delete an element from queue

```
STEP-1   If  FRONT = -1
             write  UNDERFLOW
             go to step 3
         [ end of if ]
STEP-2   If   FRONT = REAR
              FRONT = REAR= -1
         Else
              FRONT = (FRONT +1)%MAX
STEP-3   EXIT
```

PRIORITY QUEUE

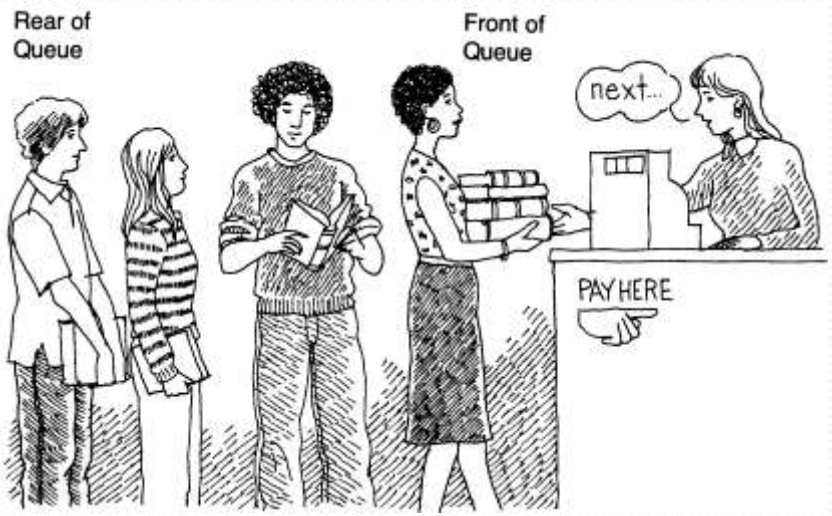
1. It is collection of elements where elements are stored according to the their priority levels.
2. Inserting and removing of elements from queue is decided by the priority of the elements.
3. An element of the higher priority is processed first.
4. Two element of same priority are processed on first-come-first-served basis.

Example: Suppose you have a few assignment from different subjects. Which assignment will you want to do first?

subjects	Due date	priority
DSGT	15 OCT	4
DLD	6 OCT	2
CYB	4 OCT	1
DS	8 OCT	3

APPLICATIONS

- Real world applications
  - Cashier line in any store.
  - Waiting on hold for tech support.
  - people on an escalator.
  - Checkout at any book store.



❖ Applications related to computer science:

1. When data is transferred asynchronously between two processes. eg. IO Buffers.
2. When a resource is shared among multiple consumers. Examples include CPU scheduling, Disk Scheduling.
3. In recognizing palindrome.
4. In shared resources management.
5. Keyboard buffer.
6. Round robin scheduling.
7. Job scheduling.
8. Simulation

THANK YOU