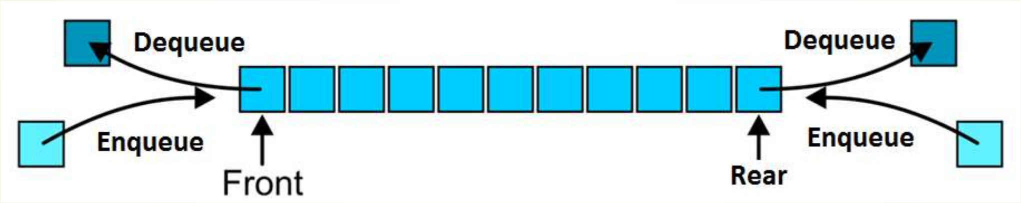
Double-Ended Queue

Double-Ended Queue (Deque)

- A Deque or deck is a double-ended queue.
- Allows elements to be added or removed on either the ends.



TYPES OF DEQUE

❑ Input restricted Deque

- Elements can be inserted only at one end.
- Elements can be removed from both the ends.

❑ Output restricted Deque

- Elements can be removed only at one end.
- Elements can be inserted from both the ends.

Deque as Stack and Queue

As STACK

- When insertion and deletion is made at the same side.

As Queue

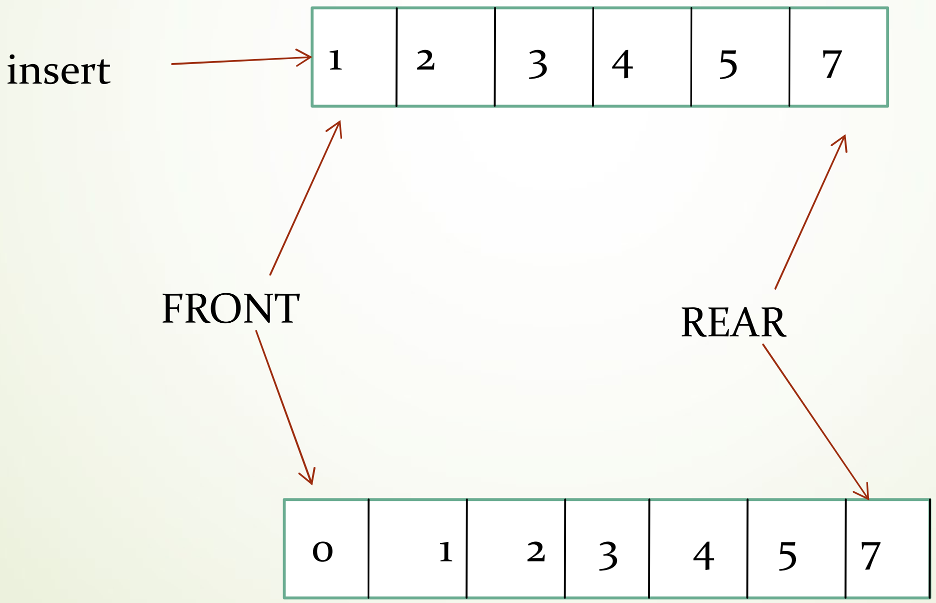
- When items are inserted at one end and removed at the other end.

OPERATIONS IN DEQUE

- Insert element at Rear
- Insert element at Front
- Delete element from Front
- Delete element from Rear

Insert_front

- insert_front() is a operation used to push an element into the front of the *Deque*.



insert

1	2	3	4	5	7

FRONT

REAR

0	1	2	3	4	5	7

Algorithm Insert_front

step1. Start

step2. Check the queue is full or not as if (r == max-1) &&(f==0)

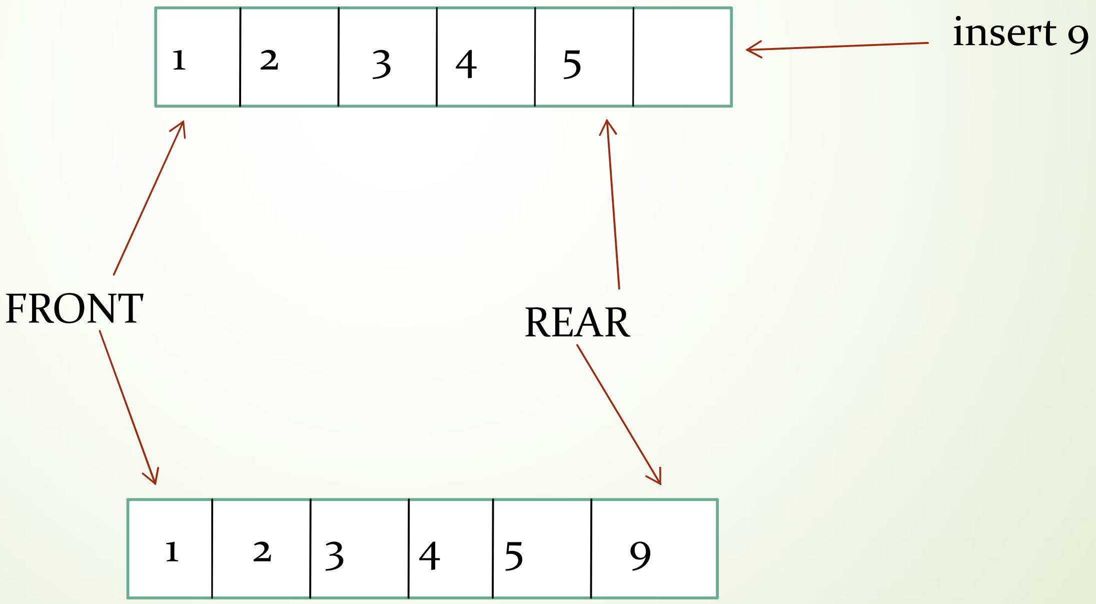
step3. If false update the pointer f as f= f-1

step4. Insert the element at pointer f as Q[f] = element

step5. Stop

Insert_back

- insert_back() is a operation used to push an element at the back of a *Deque*.



1	2	3	4	5	

insert 9

FRONT

REAR

1	2	3	4	5	9

Alogrithm insert_back

Step1: Start

Step2: Check the queue is full or not as if (r == max-1) &&(f==0) if yes queue is full

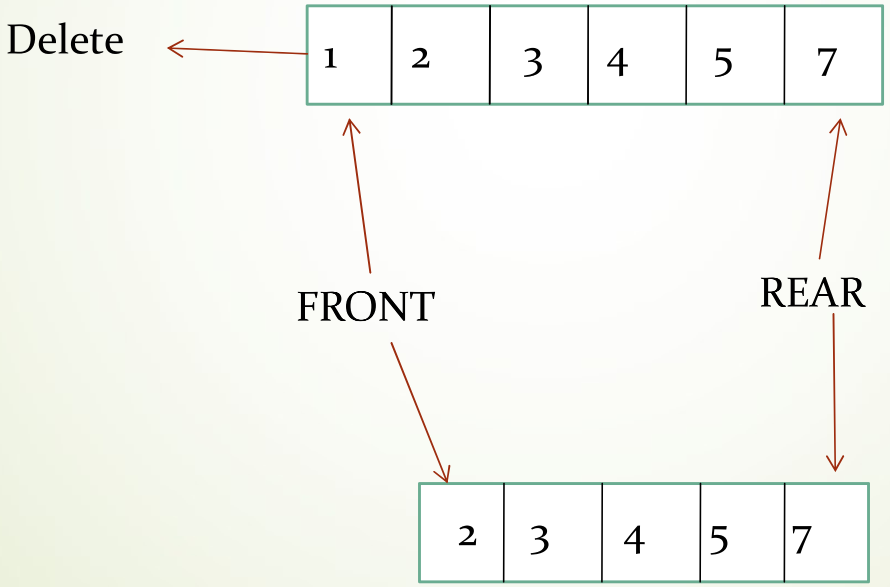
Step3: If false update the pointer r as r= r+1

Step4: Insert the element at pointer r as Q[r] = element

Step5: Stop

Delete_front

- remove_front() is a operation used to pop an element on front of the *Deque*.

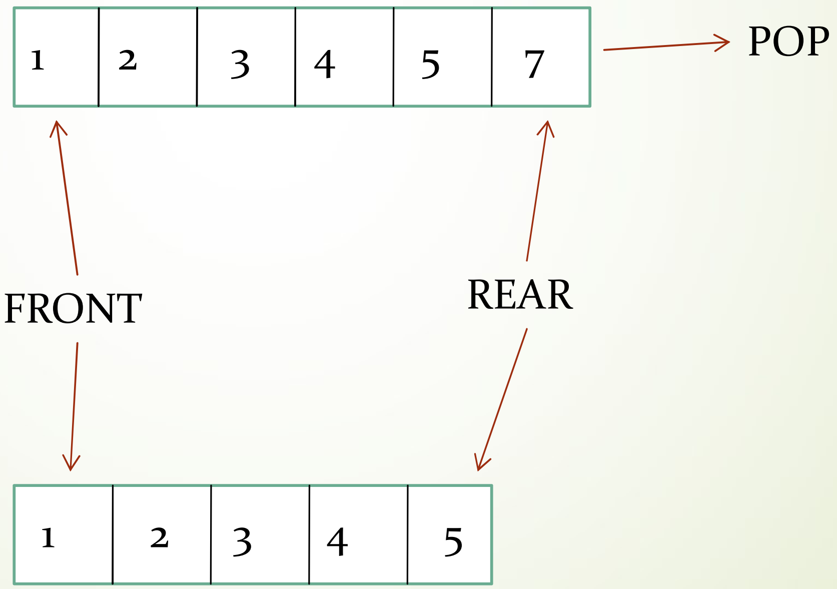


Alogrithm Delete_front

```
Step1:  Start
Step2:  Check the queue is empty or not as if (f == r and f== - 1)
                    if True  queue is empty.
Step3:  If false
    element = Q[f]
Step4:  If ( f== r) reset pointer f and r as f = r = -1
            Else update pointer f as f = f+1
Step5:  Stop
```

Remove_back

- remove_front() is a operation used to pop an element on front of the *Deque*.



1	2	3	4	5	7

1	2	3	4	5

Alogrithm Remove_back

step1. Start

step2. Check the queue is empty or not as if (f ==r&&r==-1) if yes queue is empty

step3. If false delete element at position r as element = Q[r]

step4. Update pointer r as r = r-1

step5. If (f == r) reset pointer f and r as f = r= -1

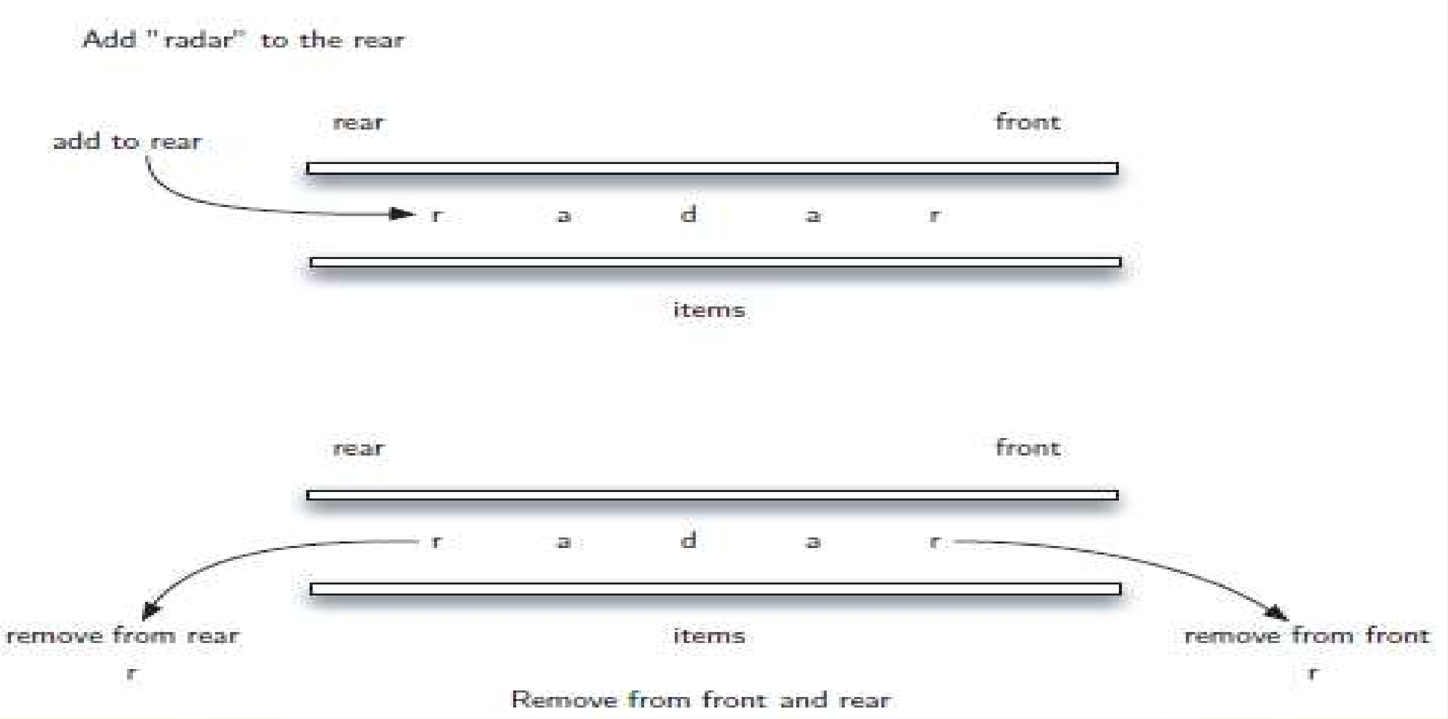
step6. Stop

Empty

- It is used to test weather the Deque is empty or not.

APPLICATIONS OF DEQUE

Palindrome-checker



APPLICATIONS OF DEQUE

A-Steal job scheduling algorithm

- The A-Steal algorithm implements task scheduling for several processors(multiprocessor scheduling).
- The processor gets the first element from the deque.
- When one of the processor completes execution of its own threads it can steal a thread from another processor.
- It gets the last element from the deque of another processor and executes it.

OTHER APPLICATIONS OF DEQUE

- **Undo-Redo** operations in Software applications.

Thank You