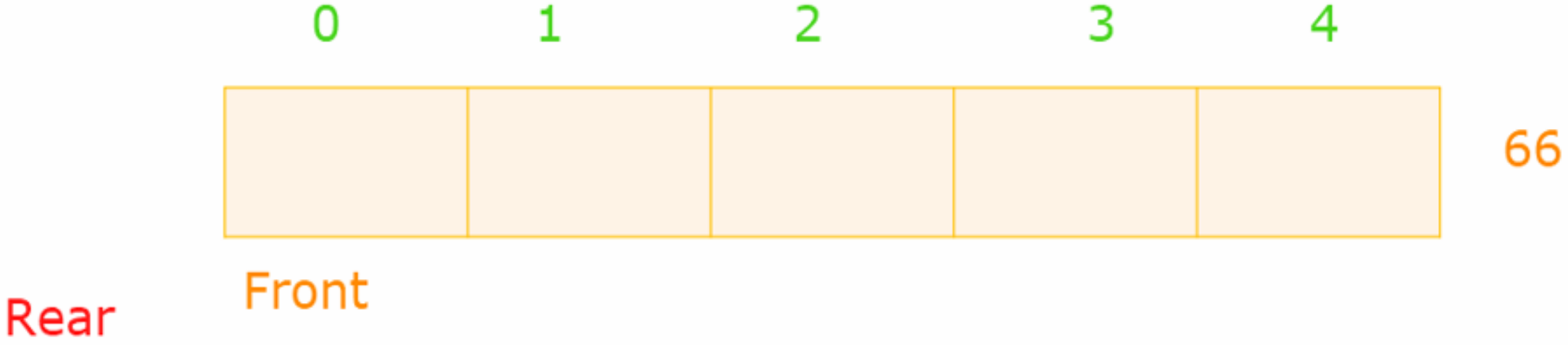
Queue

Kiran Waghmare

Queue

- Ordered collection of homogeneous elements.
- Non-primitive linear data structure.
- A new element is added at one end called rear end and the existing elements are deleted from the other end called front end.
- This mechanism is called First-In-First-Out (FIFO)
- Total no. of elements in queue = rear-front+1

Queue Operations



Insert: 1,2,3,4,5
Deletion:1,2, 3, 4,5

```
}
public static void mai
{
    Queue1 q1 = new Que
    q1.enqueue(11);
    q1.enqueue(12);
    q1.enqueue(13);
    q1.enqueue(14);
    q1.enqueue(15);
    q1.enqueue(15);
    q1.enqueue(16);
    q1.display();
    q1.dequeue();
    q1.display();
    q1.enqueue(16);
```

```
14
15

D:\Test>javac Queue1.java

D:\Test>java Queue1
11 Inserted.
12 Inserted.
13 Inserted.
14 Inserted.
15 Inserted.
Queue is full
Queue is full
11
12
13
14
15
11Deleted.
12
13
14
15
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out
        at Queue1.enqueue(Queue1.java:39)
        at Queue1.main(Queue1.java:92)
```

Representation of Queue

Queue as a data structure can be represented in two ways.

- Stack as an Array (Most popular)
- Stack as a Linked List.

Applications and uses for Queues

- **Heavily used in almost all applications of the operating system, to schedule processes, moving them in or out of process scheduler.**

- **FCFS, SJF etc**

- **Asynchronously i.e. when data resource may be the same but not received at the same rate.**

- **Anything that has to do with process and schedule, in the system or code.**

3 states of the queue

1. Queue is empty

   FRONT=REAR

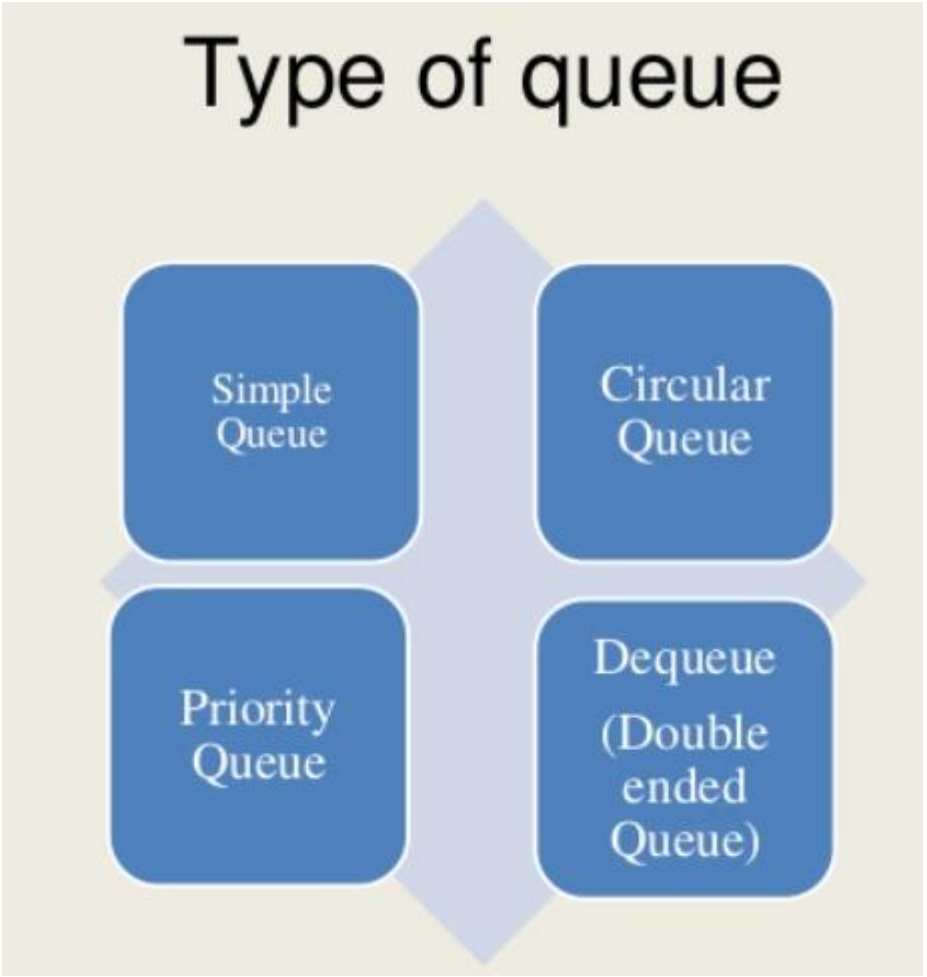
2. Queue is full

   REAR=N

3. Queue contains element >=1

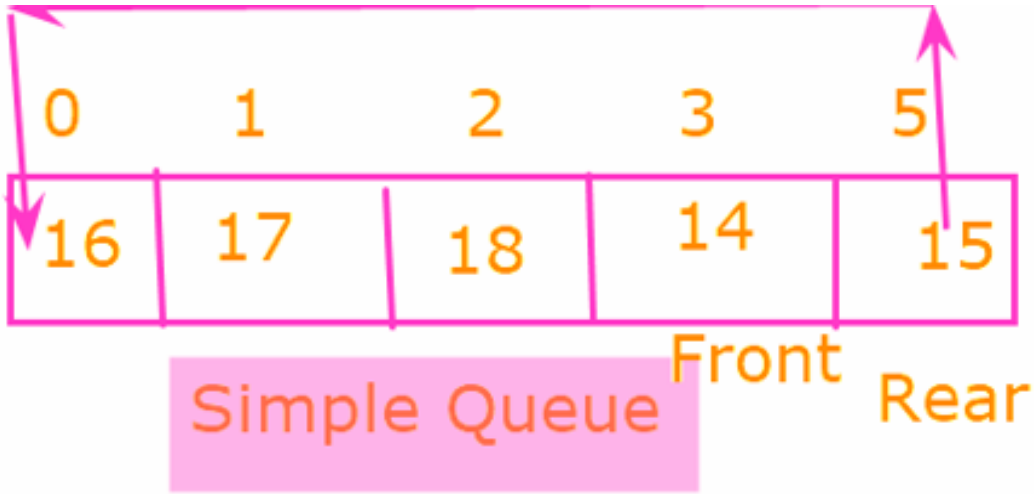
   FRONT<REAR

   NO. OF ELEMENT=REAR-FRONT+1



Types of Queue

1. Simple Queue
2. Circular Queue
3. Priority Queue
4. Deque



Simple Queue

Represent Queue in following ways:

1. Array Implementation
2. List Implementation

Circular Queue

- In a circular queue, all nodes are treated as circular. Last node is connected back to the first node.
- Circular queue is also called as **Ring Buffer.**
- It is an abstract data type.
- Circular queue contains a collection of data which allows insertion of data at the end of the queue and deletion of data at the beginning of the queue

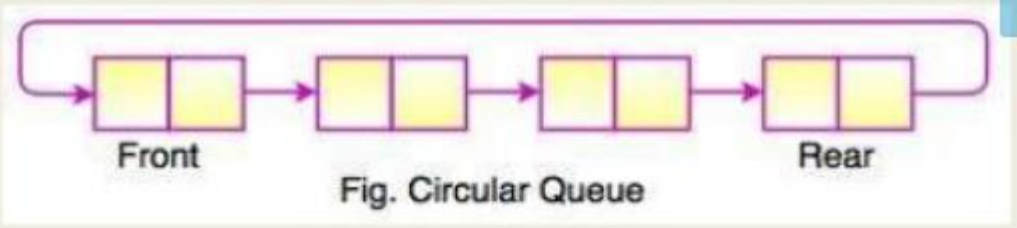


Fig. Circular Queue

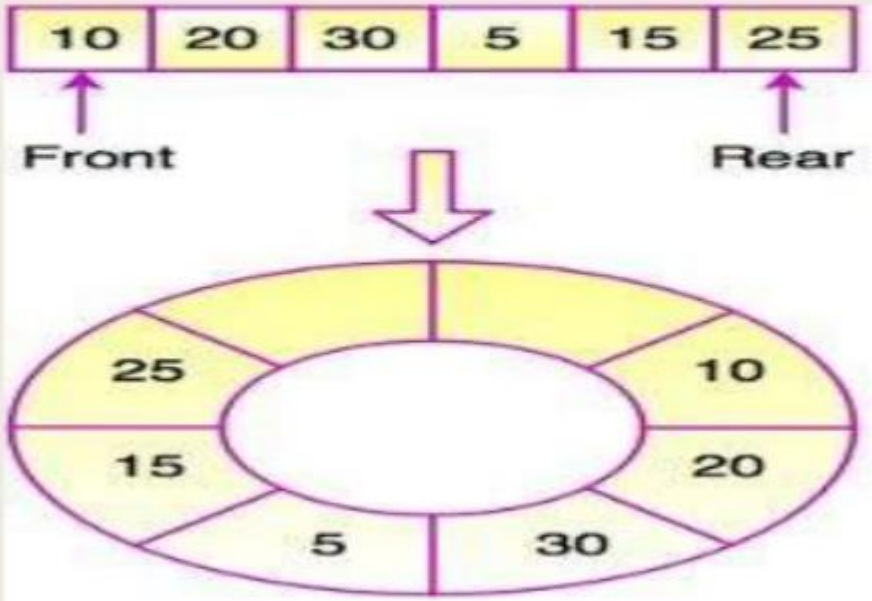
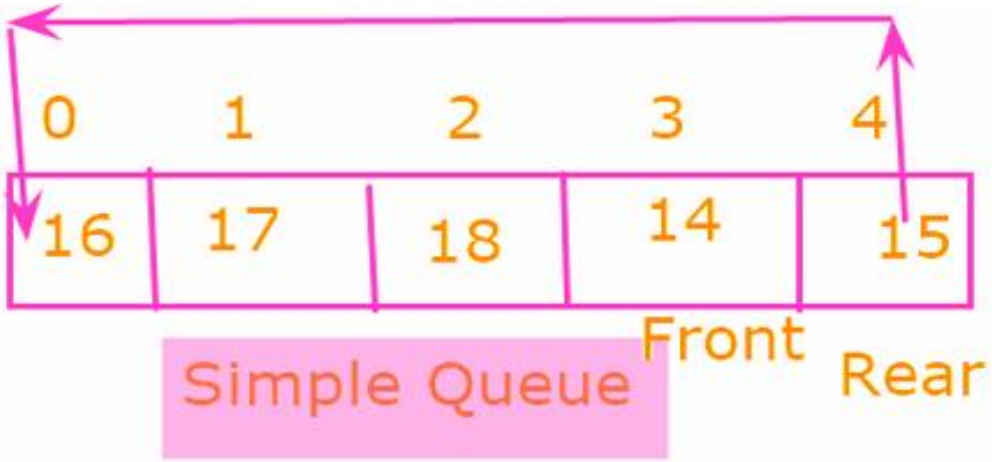


Fig. Circular Queue

Types of Queue

1. Simple Queue
2. Circular Queue
3. Priority Queue
4. Deque

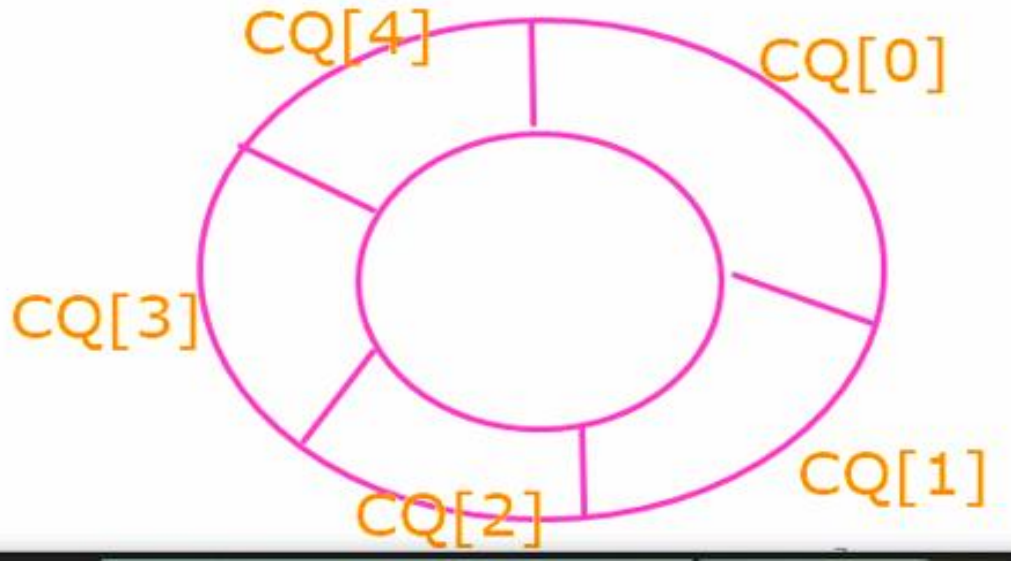


Simple Queue

size: 5 (0-4)

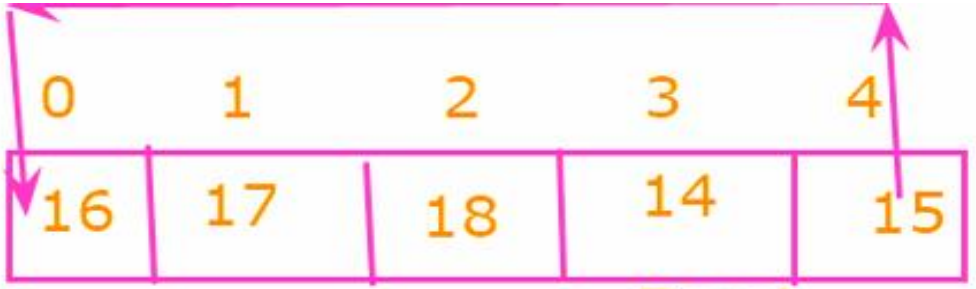
Represent Queue in following ways:

1. Array Implementation
2. List Implementation



Types of Queue

1. Simple Queue
2. Circular Queue
3. Priority Queue
4. Deque



Simple Queue

Represent Queue in following ways:

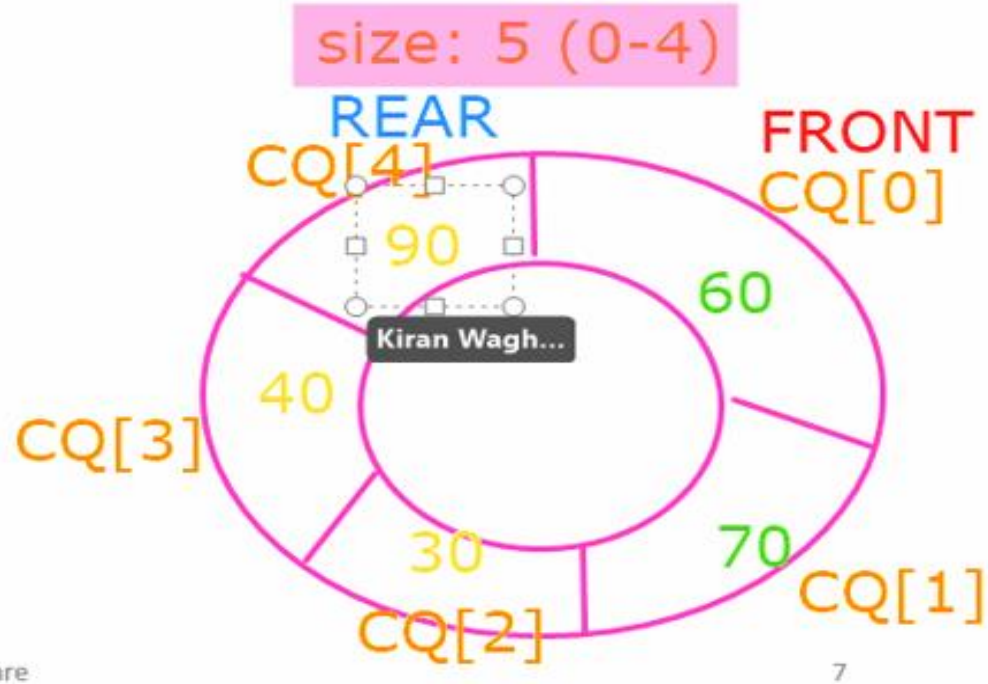
1. Array Implementation
2. List Implementation

FRONT=(FRONT+1)%SIZE

REAR= (REAR+1)%SIZE

=(4+1)%5

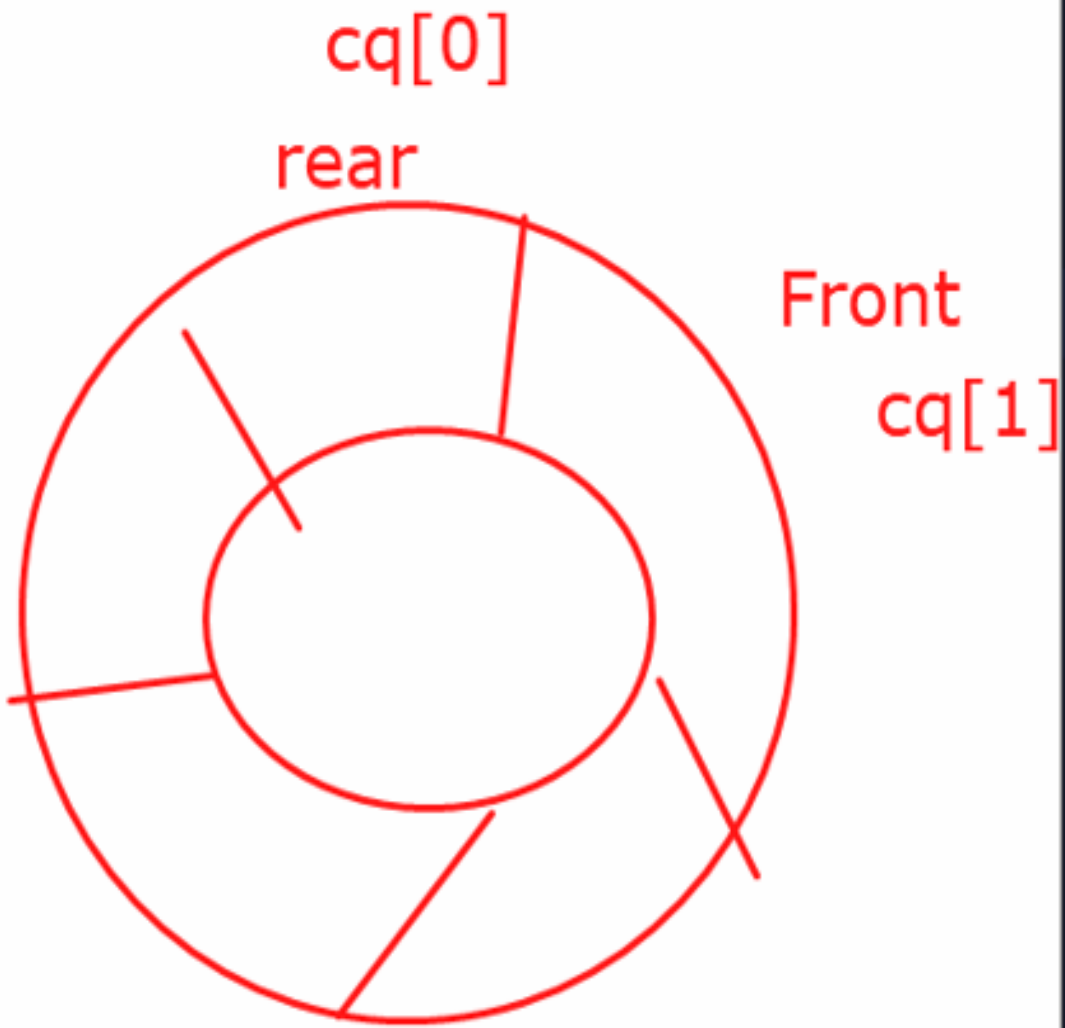
=5%5 = 0



CDAC Mumbai: Kiran Waghmare 7

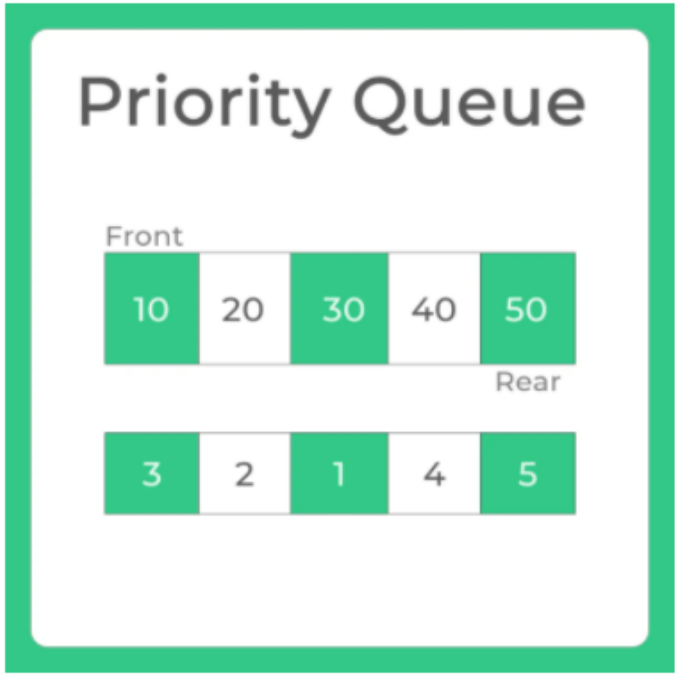
Case 1:
Front == 0
Rear = size-1

case 2:
front= rear + 1



Priority Queue

Priority Queue is a special type of queue in which elements are treated according to their priority. Insertion of an element take place at the rear of the queue but the deletion or removal of an element take place according to the priority of the element. Element with the highest priority is removed first and element with the lowest priority is removed last.



Applications of Priority Queue

The applications of priority queue are:-

- Dijkstra's shortest path algorithm
- Data compression in huffman codes.
- Load balancing and interrupt handling in operating system.
- Sorting heap.

Priority Queue

DST : Heap

-Min → Min Priority Queue

-Max → Max Priority Queue

E.g: Min heap, Max heap

Delete

Insertion

Priority : Max / Min

Operation

1.Insert

2.Delete

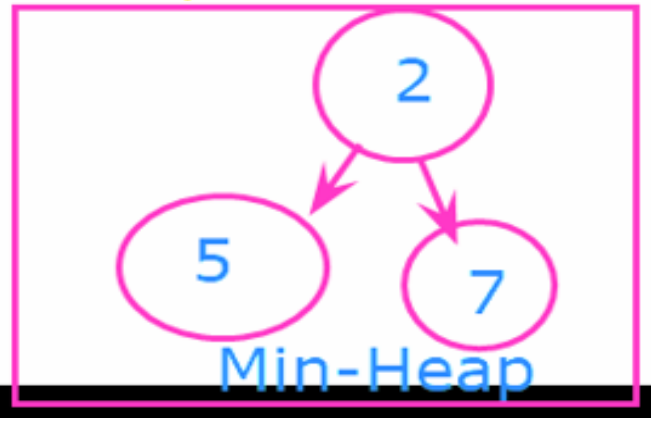
3.Find/Search

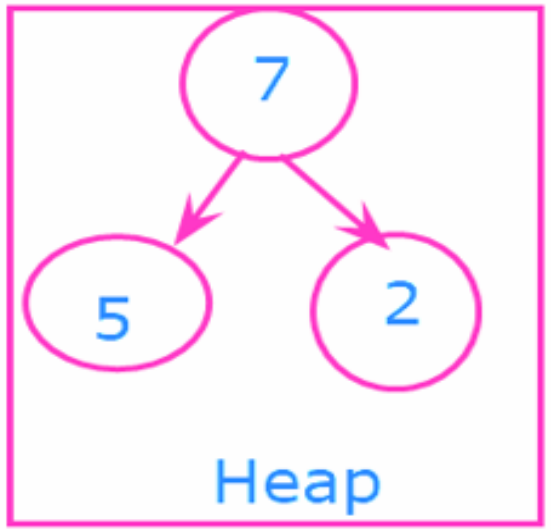
Delete : dequeue()

Insert : 2 4 7 3 9 1 4

Max :priority

9 7 4 4 3 2 1





Max-Heap

CDAC Mumbai: Kiran Waghmare

Dequeue
(Double ended Queue)

- In Double Ended Queue, insert and delete operation can be occur at both ends that is front and rear of the queue

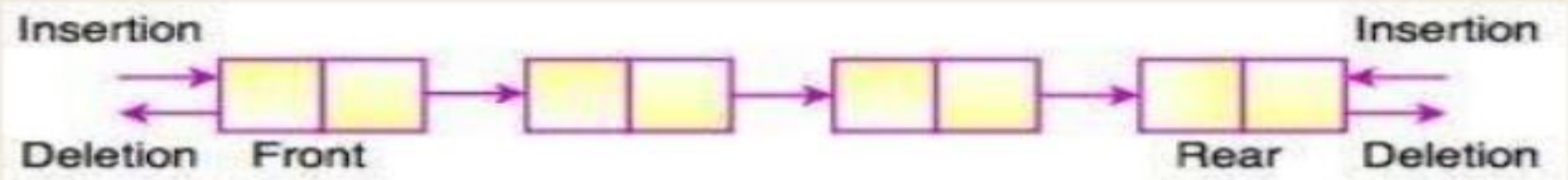


Fig. Double Ended Queue (Dequeue)

Deque: Double Ended Queue

First In First Out

Types of Deque:

1. Input Retricted Deque
   - -Input: one end
   - -Deletion: both end
2. Output Retricted Deque
   - -Input:both end
   - -Deletion: one end

Operations:

- insertFront()
- insertRear()

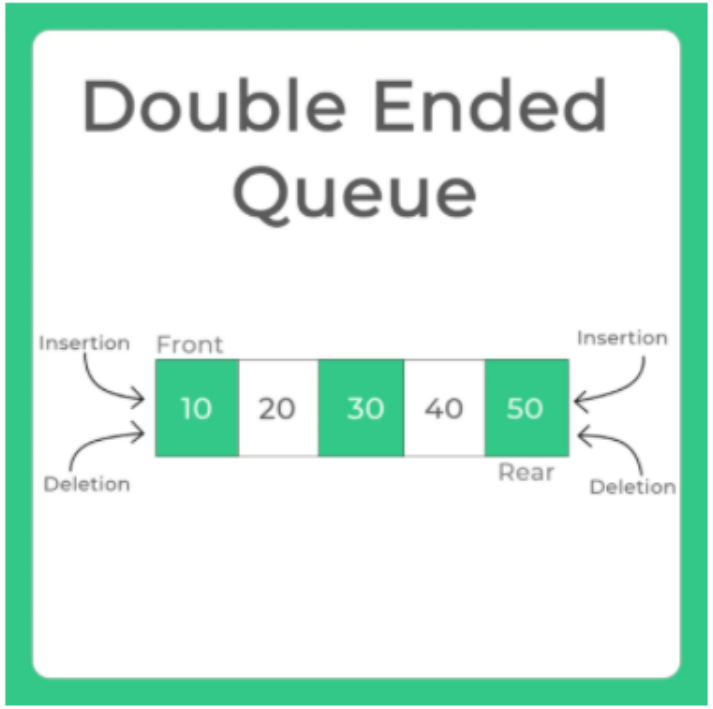
→ enqueu

- deleteFront()
- deleteRear()

→ dequeue

- isFull()
- isEmpty()
- display()

Double Ended Queue



Double ended queue are also known as deque. In this type of queue insertion and deletion of an element can take place at both the ends. Further deque is divided into two types:-

- Input Restricted Deque :- In this, input is blocked at a single end but allows deletion at both the ends.
- Output Restricted Deque :- In this, output is blocked at a single end but allows insertion at both the ends.

Applications of Double Ended Queue

The applications of double ended queue are:-

- To execute undo and redo operation.
- For implementing stacks.
- Storing the history of web browsers.

Thanks