Queue

1. Queue is a linear data structure that is referred as First In First Out list.

1. Queue Operation
2. Enqueue: Add an element to the end of the queue
3. Dequeue: Remove an element from the front of the queue

IsEmpty: Check if the queue is empty

IsFull: Check if the queue is full

Peek: Get the value of the front of the queue without removing it

There are two ways of implementing the Queue:

Sequential allocation: The sequential allocation in a Queue can be implemented using an array.  
For more details, click on the below link:<https://www.javatpoint.com/array-representation-of-queue>

Linked list allocation: The linked list allocation in a Queue can be implemented using a linked list.  
For more details, click on the below link: <https://www.javatpoint.com/linked-list-implementation-of-queue>

### Types of Queue

1. Simple Queue/linear Queue

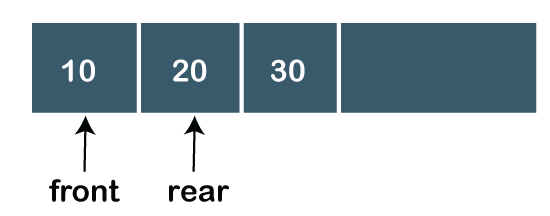
2. Circular Queue

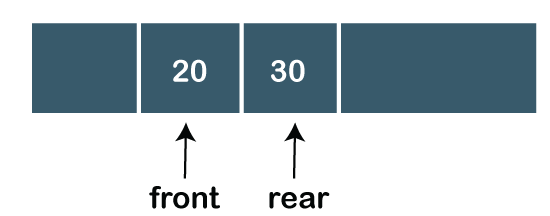
3. Priority Queue

4. Double Ended Queue

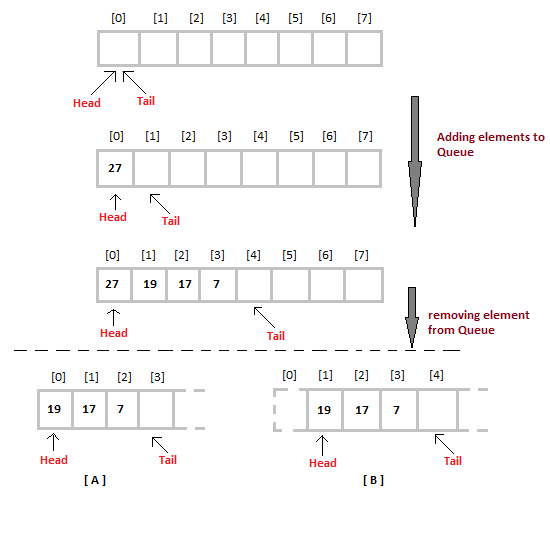
**Simple Queue**

In Linear Queue, an insertion takes place from one end while the deletion occurs from another end. The end at which the insertion takes place is known as the rear end, and the end at which the deletion takes place is known as front end. It strictly follows the FIFO rule.





**Implemention of Queue**



**Drawback of linear queue**

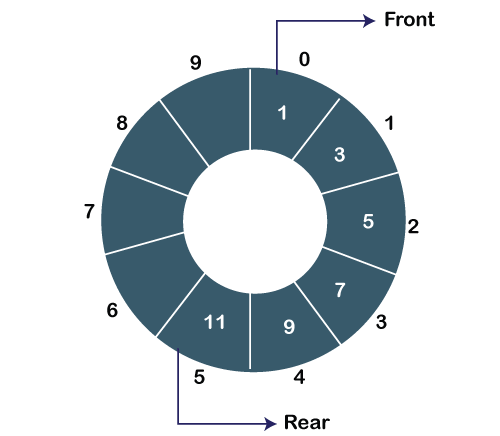
The major drawback of using a linear Queue is that insertion is done only from the rear end. If the first three elements are deleted from the Queue, we cannot insert more elements even though the space is available in a Linear Queue. In this case, the linear Queue shows the overflow condition as the rear is pointing to the last element of the Queue.

The only way is to reset the linear queue, for a fresh start.

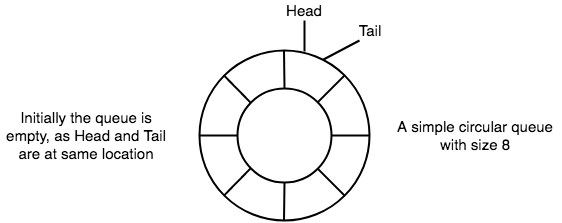
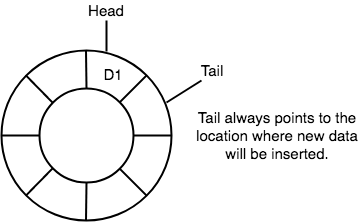
**Circular Queue**

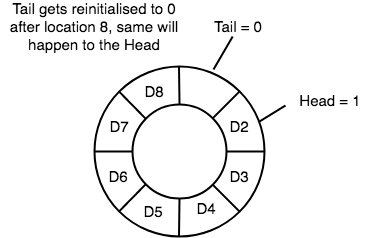
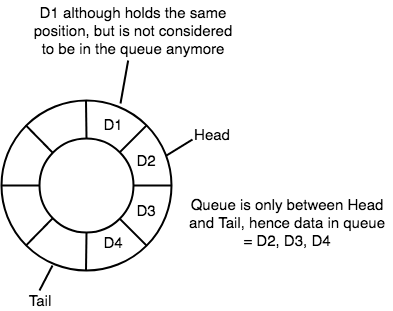
In Circular Queue, all the nodes are represented as circular. It is similar to the linear Queue except that the last element of the queue is connected to the first element. It is also known as Ring Buffer as all the ends are connected to another end

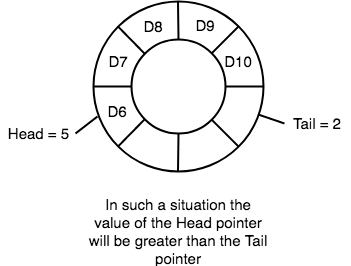
In circular queue, the problem is solved that is faced in linear queue.



**Implementation of Circular Queue**





https://www.youtube.com/watch?v=ArVoZ1KrHDY

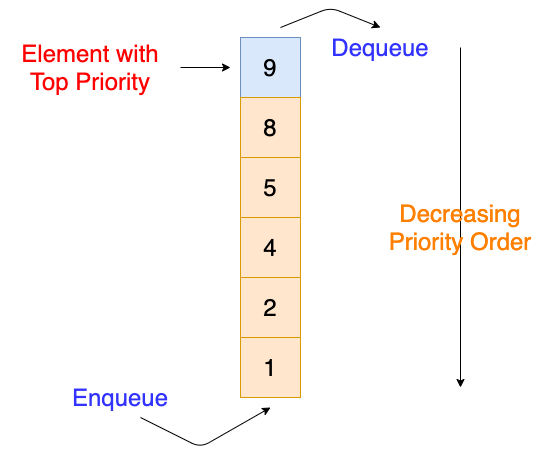
**Priority Queue**

A priority queue is another special type of Queue data structure in which each element has some priority associated with it. Based on the priority of the element, the elements are arranged in a priority queue. If the elements occur with the same priority, then they are served according to the FIFO principle.

**There are two type of priority queue**

1. Ascending priority queue
2. Descending priority queue

**Priority Queues** can be **implemented** using common data structures like arrays, linked-lists, heaps and binary trees.



**Applications:**

* Cup Scheduling
* Graph Algorithms (Dijkstra shortest path, Prim’s minimum spanning tree)
* [Data compression](https://en.wikipedia.org/wiki/Data_compression)
* [Heap Sort](https://www.geeksforgeeks.org/heap-sort/)
* Involving priority there will be implemented

**Double Ended Queue**

Dequeue or Double Ended Queue is a generalized version of Queue data structure that allows insert and delete at both ends.

