**Priority Queue**

**Priority Queue**is an [abstract data type](https://www.geeksforgeeks.org/abstract-data-types/) that is similar to a [queue](https://www.geeksforgeeks.org/queue-data-structure/), and every element has some priority value associated with it. The priority of the elements in a priority queue determines the order in which elements are served (i.e., the order in which they are removed). If in any case the elements have same priority, they are served as per their ordering in the queue.

**Properties of Priority Queue**

***So, a priority Queue is an extension of the***[***queue***](https://www.geeksforgeeks.org/queue-set-1introduction-and-array-implementation/)***with the following properties.***

* Every item has a priority associated with it.
* An element with high priority is dequeued before an element with low priority.
* If two elements have the same priority, they are served according to their order in the queue.

## Types of Priority Queue:

### **1) Ascending Order Priority Queue**

As the name suggests, in ascending order priority queue, the element with a lower priority value is given a higher priority in the priority list.

### **2) Descending Order Priority Queue**

A descending order priority queue gives the highest priority to the highest number in that queue

**Dequeue**

[Deque or Double Ended Queue](http://en.wikipedia.org/wiki/Double-ended_queue) is a generalized version of [Queue data structure](https://www.geeksforgeeks.org/queue-set-1introduction-and-array-implementation/)that allows insert and delete at both ends.

**Operations on Deque:** Mainly the following four basic operations are performed on queue:  
***insertFront()***: Adds an item at the front of Deque.  
***insertLast()***: Adds an item at the rear of Deque.  
***deleteFront()***: Deletes an item from the front of Deque.  
***deleteLast()***: Deletes an item from the rear of Deque. In addition to the above operations, the following operations are also supported.  
***getFront()***: Gets the front item from the queue.  
***getRear()***: Gets the last item from queue.  
***isEmpty()***: Checks whether Deque is empty or not.  
***isFull()***: Checks whether Deque is full or not.

**Some Practical Applications of Deque**:

* Applied as both stack and queue, as it supports both operations.
* Storing a web browser’s history.
* Storing a software application’s list of undo operations.
* Job scheduling algorithm