**Queue**

* Queue is FIFO, i.e., whichever element comes first goes out first.
* Adding an item=Enqueue()
* Deleting an item =Dequeue()
* Need two variables for data structure front- for deletion, and Back- for insertion
* Variations: Dequeue, Priority queue and doubly ended priority queue.

**Applications:**

* Single resource and multiple consumers to be served in arrival order, e.g., Doctor-Patients.
* Synchronization between slow and fast processes/devices, e.g., Keyboard-Processor.
* In Operating system:
  + FCFS scheduling.
  + Semaphores for sleeping processes.
  + Buffer.
  + Spooling in printers.
* In network:
  + In routers /Switches for buffer.
  + Mail queues

**Java Implementations**: Queue is an interface in java.

Collections<-Queue<-LinkedList(Doubly Linked list implementations)

Collections<-Queue<-ArrayDeque(Circular Array impl)

Functions: offer(x),peek(),poll();

Similar function which throw exceptions: Add(),element(),remove();

All functions have O(1) complexity.

**Dequeue**: Queue with insertion and deletion supported at both ends.

Applications:

* It can be used as stack and queue both.
* Used to maintain a history of action when we have limited buffer.
* Steal process scheduling algorithm in multi-processor env.
* Implement priority queue with only two priorities. Higher priority is inserted at front while lower priority is inserted at end. Deletion is always at front.
* Max/min of all subarrays of size k in given array.

Dequeu.descendingIterator() for iterating through last to first.