Queue

Like Stack, Queue is a linear structure which follows a particular order in which the operations are performed. The order is First In First Out (FIFO).

In a Queue data structure, we maintain two pointers, front and rear. The front points the first item of queue and rear points to last item. Insertion occurs from rear of the queue and deletion from the front of the queue.

A good example of queue is any queue of consumers for a resource where the consumer that came first is served first.
The difference between stacks and queues is in removing. In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.

Operations on Queue

Mainly the following four basic operations are performed on queue:

1. Enqueue: Adds an item to the queue. If the queue is full, then it is said to be an Overflow condition.
2. Dequeue: Removes an item from the queue. The items are popped in the same order in which they are pushed. If the queue is empty, then it is said to be an Underflow condition.
3. Front: Get the front item from queue.
4. Rear: Get the last item from queue.

Applications of Queue:
Queue is used when things don't have to be processed immediately, but have to be processed in First InFirst Out order like Breadth First Search. This property of Queue makes it also useful in following kind of scenarios.

1. When a resource is shared among multiple consumers. Examples include CPU scheduling, Disk Scheduling.
2. When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes. Examples include IO Buffers, pipes, file IO, etc.

Implementation:

- *Array implementation Of Queue:*
  For implementing queue, we need to keep track of two indices, front and rear. We enqueue an

item at the rear and dequeue an item from front. If we simply increment front and rear indices, then there may be problems, front may reach end of the array. The solution to this problem is to increase front and rear in circular manner

- *Linked List implementation of Queue*
  See code

Additional reading

- [Queue using Stacks](#)