**Queue Data Structure**

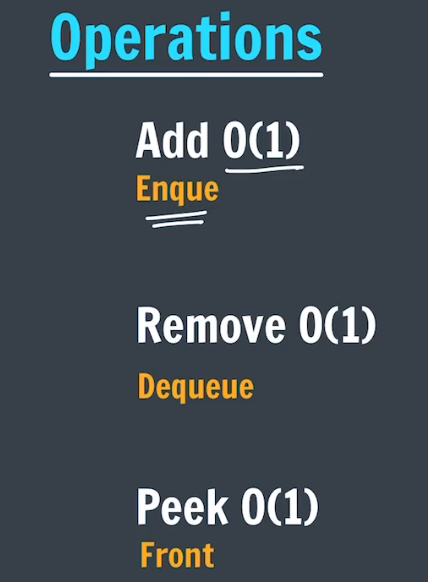
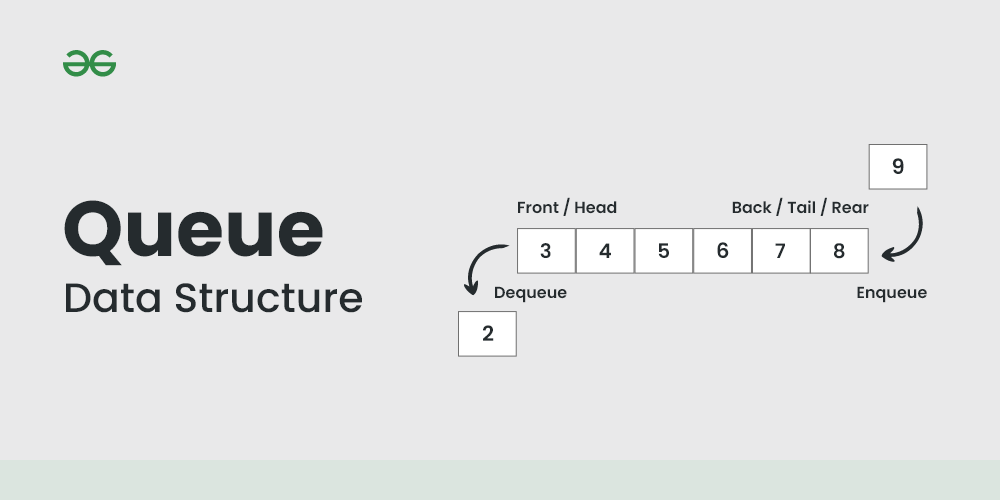
**What is Queue in Data Structures?**

A queue is a linear data structure that follows the**First-In-First-Out (FIFO)** principle. It operates like a line where elements are added at one end (**rear**) and removed from the other end (**front**).

**Basic Operations of Queue Data Structure**

* **Enqueue (Insert)**: Adds an element to the rear of the queue.
* **Dequeue (Delete)**: Removes and returns the element from the front of the queue.
* **Peek**: Returns the element at the front of the queue without removing it.
* **Empty**: Checks if the queue is empty.
* **Full**: Checks if the queue is full.

**Linear Queue**



* **Time Complexity:** O(1) for Enqueue (element insertion in the queue) as we simply increment pointer and put value in array and O(N) for Dequeue (element removing from the queue) as we use for loop to implement that.

**Auxiliary Space:**O(N), as here we are using an N size array for implementing Queue

We can notice that the Dequeue operation is O(N) which is not acceptable. The enqueue and dequeue both operations should have O(1) time complexity.

* That is why if we wish to implement a queue using array, we do[circular array implementation of queue](https://www.geeksforgeeks.org/circular-queue-set-1-introduction-array-implementation/). Here both enqueue and dequeue takes O(1) time complexity. Though, **Auxiliary Space:**O(N), as here we are using an N size array for implementing circular Queue.
* For linear queue to be implemented (for both enqueue & dequeue) in O(1), it has to be implemented using linked list.