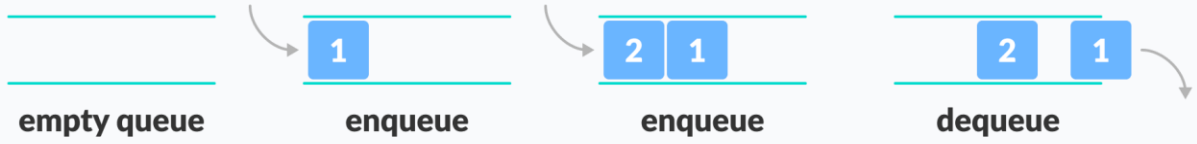
Queue Data Structure

In this tutorial, you will learn what a queue is. Also, you will find implementation of queue in C, C++, Java and Python.

A queue is a useful data structure in programming. It is similar to the ticket queue outside a cinema hall, where the first person entering the queue is the first person who gets the ticket.

Queue follows the **First In First Out (FIFO)** rule - the item that goes in first is the item that comes out first.



FIFO Representation of Queue

In the above image, since 1 was kept in the queue before 2, it is the first to be removed from the queue as well. It follows the **FIFO** rule.

In programming terms, putting items in the queue is called **enqueue**, and removing items from the queue is called **dequeue**.

We can implement the queue in any programming language like C, C++, Java, Python or C#, but the specification is pretty much the same.

Basic Operations of Queue

A queue is an object (an abstract data structure - ADT) that allows the following operations:

- **Enqueue**: Add an element to the end of the queue
- **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

Working of Queue

Queue operations work as follows:

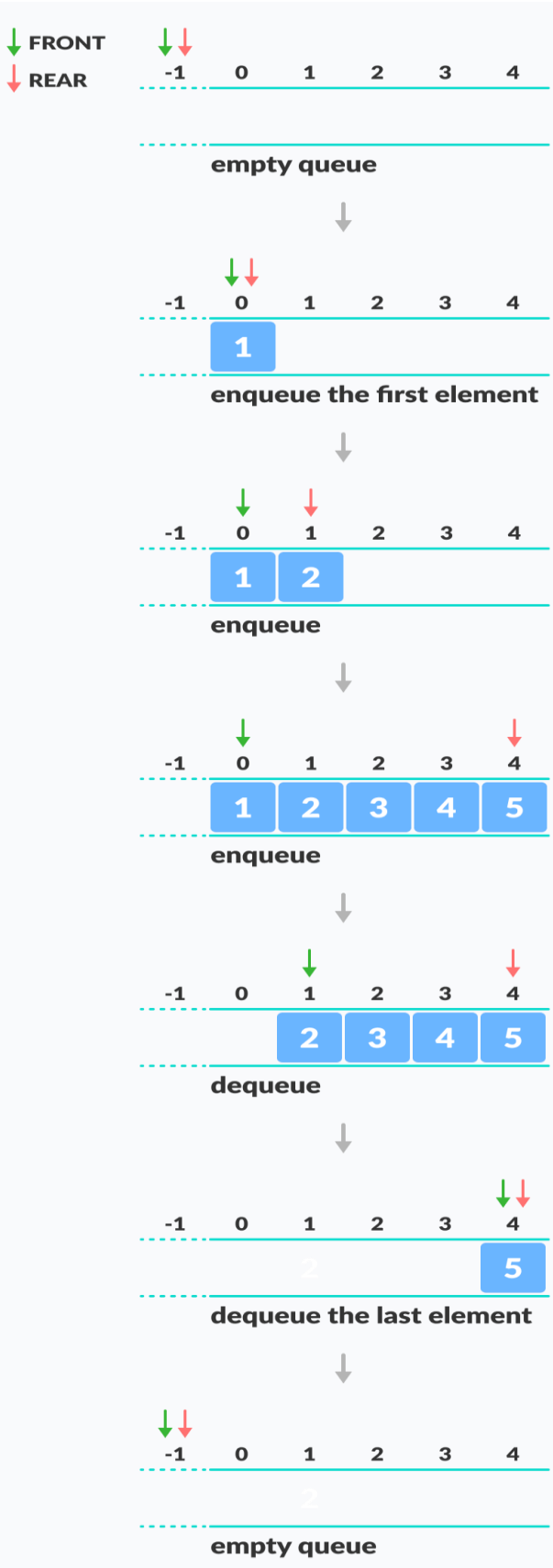
- two pointers `FRONT` and `REAR`
- `FRONT` track the first element of the queue
- `REAR` track the last element of the queue
- initially, set value of `FRONT` and `REAR` to -1

Enqueue Operation

- check if the queue is full
- for the first element, set the value of `FRONT` to 0
- increase the `REAR` index by 1
- add the new element in the position pointed to by `REAR`

Dequeue Operation

- check if the queue is empty
- return the value pointed by `FRONT`
- increase the `FRONT` index by 1
- for the last element, reset the values of `FRONT` and `REAR` to -1


FRONT
REAR
-1 0 1 2 3 4
empty queue
1
enqueue the first element
1 2
enqueue
1 2 3 4 5
enqueue
2 3 4 5
dequeue
5
dequeue the last element
empty queue