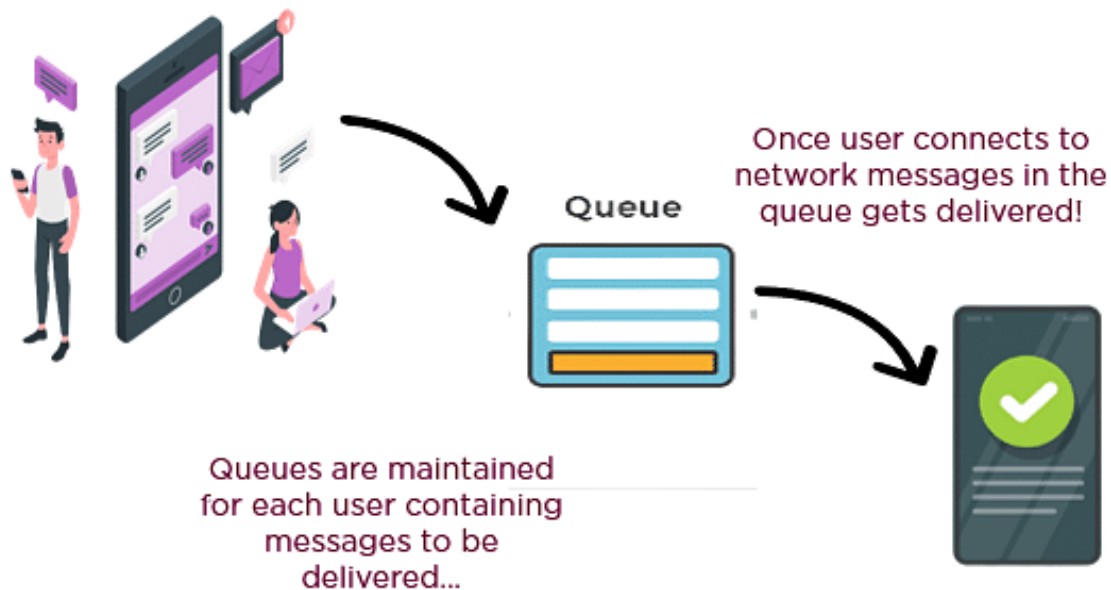# Data Structures and Algorithms

Lecture 18: Queue – Operations

Operations in Queue

The various operations that are supported by a queue data structure that helps the user to modify and manipulate the data present in the queue
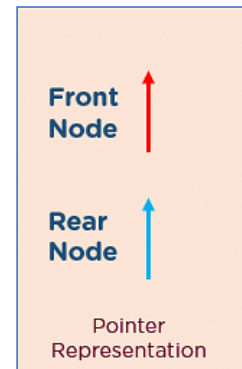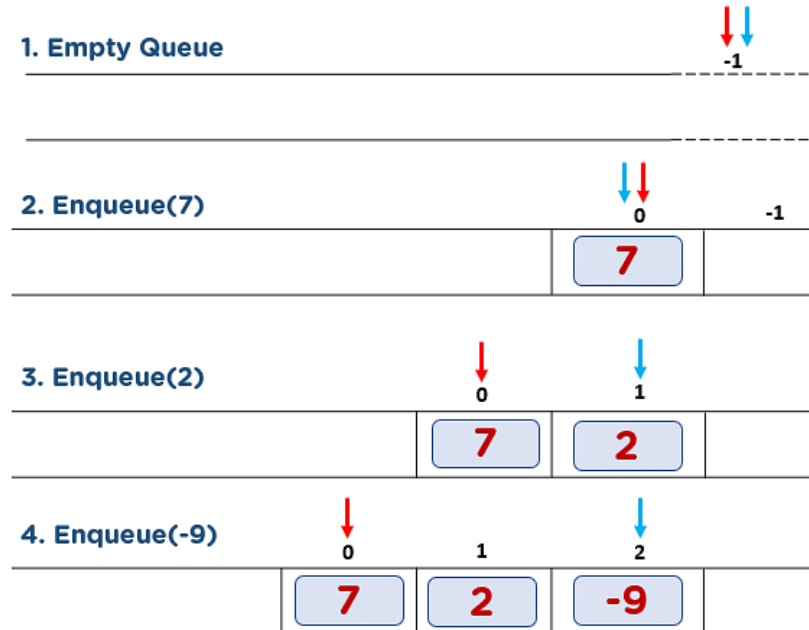

Once user connects to network messages in the queue gets delivered!

Queue

Queues are maintained for each user containing messages to be delivered...

Operations in Queue

1) Enqueue() - Insertion of elements to the queue.

2) Dequeue() - Removal of elements from the queue.

3) Peek() - Acquires the data element available at the front node of the queue without deleting it.

4) isFull() - Validates if the queue is full.

5) isNull() - Checks if the queue is empty.

Enqueue() Operation

The following steps should be followed to insert (enqueue) data element into a queue:

1) Step 1: Check if the queue is full.

2) Step 2: If the queue is full, Overflow error.

3) Step 3: If the queue is not full, increment the rear pointer to point to the next available empty space.

4) Step 4: Add the data element to the queue location where the rear is pointing.

5) Step 5: Here, you have successfully added 7, 2, and -9.

1. Empty Queue

-1

2. Enqueue(7)

0   -1

7

3. Enqueue(2)

0   1

7   2

4. Enqueue(-9)

0   1   2

7   2   -9

Front Node

Rear Node

Pointer Representation

Dequeue() Operation
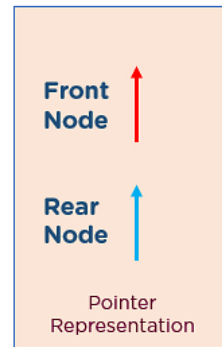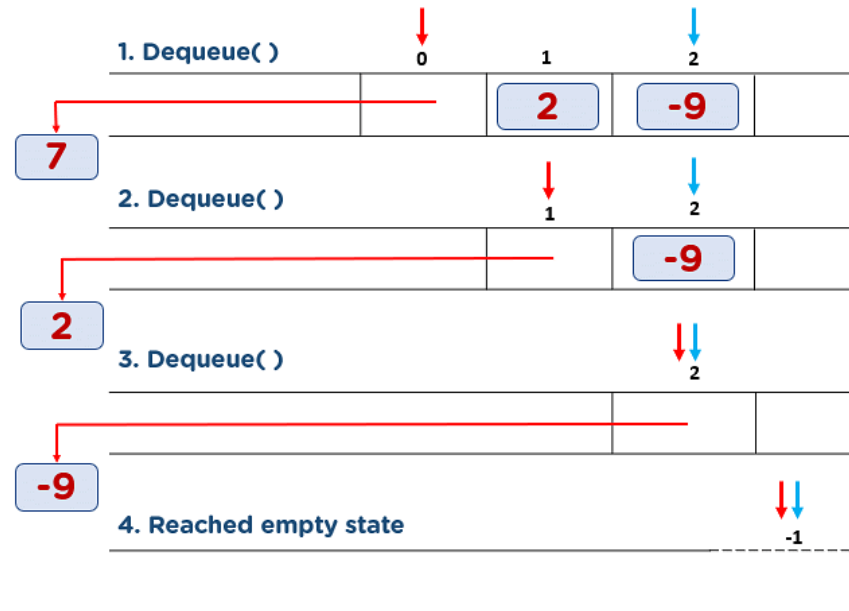
Steps to remove data from the queue -

Step 1: Check if the queue is empty.

Step 2: If the queue is empty, Underflow error.

Step 3: If the queue is not empty, access the data where the front pointer is pointing.

Step 4: Increment front pointer to point to the next available data element.

Step 5: Here, you have removed 7, 2, and -9 from the queue data structure.

Peek() Operation

This function helps in extracting the data element where the front is pointing without removing it from the queue. The algorithm of Peek() function is as follows-

Step 1: Check if the queue is empty.

Step 2: If the queue is empty, return "Queue is Empty."

Step 3: If the queue is not empty, access the data where the front pointer is pointing.

Step 4: Return data.

isFull() Operation

This function checks if the rear pointer is reached at MAXSIZE to determine that the queue is full. The following steps are performed in the isFull() operation -

Step 1: Check if rear == MAXSIZE - 1.

Step 2: If they are equal, return "Queue is Full."

Step 3: If they are not equal, return "Queue is not Full."

isNull() Operation

The algorithm of the isNull() operation is as follows -

Step 1: Check if the rear and front are pointing to null memory space, i.e., -1.

Step 2: If they are pointing to -1, return "Queue is empty."

Step 3: If they are not equal, return "Queue is not empty."

Recall:

- A queue is an object which is used to manipulate the ordered collection of different data types. The various queue operations like Enqueue(), Dequeue(), isFull(), isNull() and Peek() are operated on Queue. It is recommended to apply queue when there is a need for the FCFS (First Come First Serve) approach in software development.