DAILY DSA | DAY-16 | Queues|
-GOPALKRISHNA A

The word "queues" is a common word that we use & practice in real time, starting from vehicle toll booths to Gopalan malls!

The same principle applies in the data arrangement/ordering when using queues.

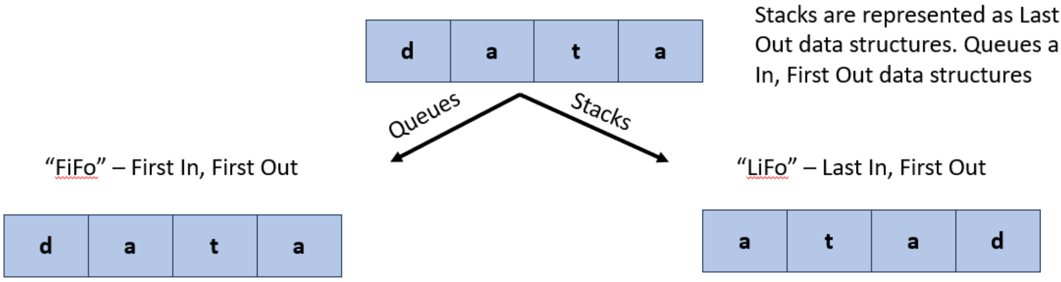
Queue: A queue is a linear data structure that stores the data that follows the **First in First Out (FIFO)** order that is open at both ends.

- When an item is inserted into a queue, it's called **"enqueued"**
- When an item is removed, it is **"dequeued"**

We define a queue to be a list in which all additions to the list are made at one end, and all the deletions from the list are made at the other end. The element that is first pushed into the order, the operation is first performed on that.

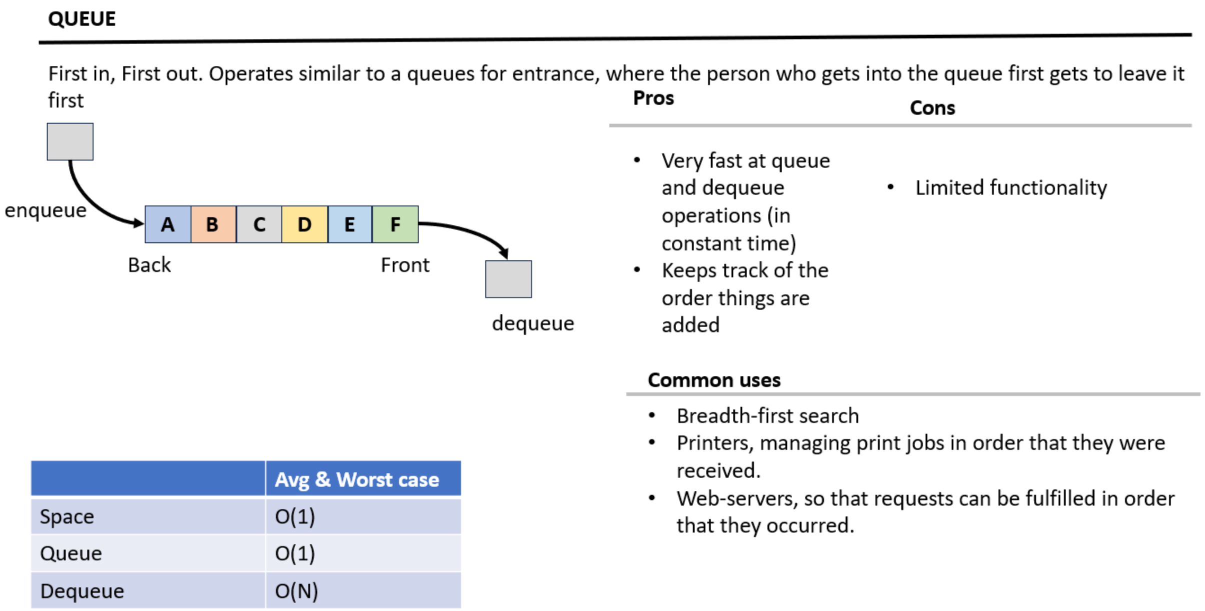
Q: Why do we need a different method of processing data? When we have stacks that work on the principle of LIFO?

A: **In queues, items are processed in the order they are inserted.** Suppose we have a stream of data in the node. As it comes in, we need to do something to it and then write it to a file to read later. For simplicity, assume we need to capitalize every letter streamed.



Stacks are represented as Last In, First Out data structures. Queues are First In, First Out data structures

The main reason is queues process data fairly and preserve the order of the collection. This also happens when we iterate over items with a for or while loop from index 0 to index.length -1



enqueue
A
B
C
D
E
F
Back
Front
dequeue

QUEUE

First in, First out. Operates similar to a queues for entrance, where the person who gets into the queue first gets to leave it first

Pros

- Very fast at queue and dequeue operations (in constant time)
- Keeps track of the order things are added

Cons

- Limited functionality

Common uses

- Breadth-first search
- Printers, managing print jobs in order that they were received.
- Web-servers, so that requests can be fulfilled in order that they occurred.

	Avg & Worst case
Space	O(1)
Queue	O(1)
Dequeue	O(N)

Main operations in the queue:

- **Enqueue**: To insert elements at the rear end of the queue
- **Dequeue**: Dequue operations perform the deletion from the front end of the queue.
- **Queue overflow**: When the queue is completely full, it shows an overflow
- **Queue underflow**: When there is no element in the queue, it shows an underflow

Types of queue:

- **Linear queue:** In linear queue, an insertion takes place from one end while deletion occurs from another end. strictly follows the FIFO rule.
- **Circular queue**: In the circular queue, all nodes are represented as circular. It is similar to a linear queue except that the last element of the queue is connected to the first element.
- **Priority queue**: The queue in which each element has some priority associated with it based on the priority of the element, elements are arranged in a priority queue. If elements occur with the same priority, then they are served according to the FIFO principle.

Advantages of Queue:

- A large amount of data can be managed efficiently with ease
- Operations such as insertion and deletion can be performed with ease as they follow the first in first out rule
- Queues are fast in speed for data inter-process communication

Disadvantages of Queue:

- The operations such as insertion and deletion of elements from the middle are time-consuming
- Limited space
- The maximum size of the queue needs to be defined prior.