

Chapter III: Stacks and Queues

1) Definition of “Queue”:

A queue is a structure that stores items in an orderly manner, but makes only one of them accessible, called the head of the queue. When an element is added, it becomes the last element that will be accessible. When an element is removed from the line, the head is always removed as it is the first element that has been placed in the line.

To summarize, the first item added to the stack is the first item removed from it. This structure is also called a **FIFO (first in, first out) list**



2) Basic Operations on a Queue:

1. **Enqueue (insert):**

Adds an element to the **rear** (or end) of the queue.

2. **Dequeue (remove):**

Removes and returns the element from the **front** of the queue.

3. **Front (or Peek):**

Returns the element at the front of the queue **without removing it**.

4. **IsEmpty:**

Checks whether the queue contains any elements.

5. **IsFull (in bounded queues):**

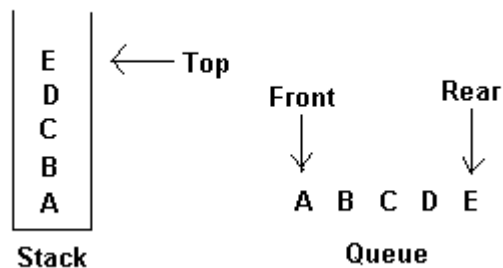
Checks whether the queue has reached its maximum capacity.

3) Some Use Cases of Queue

In computer science, a file is essentially used to store data that must be processed in the order in which it arrives.

- The best-known example of a computer queue is when a printer prints documents received by it, printing the first document that arrives and finishing with the last one.
- Managing the programs to be executed
- Managing processes waiting for a system resource.
- Simulation: Modeling of a customer queue in front of a counter.
- Messages waiting in a telephone network switch.

4) Comparing Stack and Queue Structure



Data Structure Behavior Example

- **Queue Example (FIFO):**

`enqueue(1) → enqueue(2) → enqueue(3)`

`dequeue()` → returns 1 → Next `dequeue()` returns 2.

- **Stack Example (LIFO):**

`push(1) → push(2) → push(3)`

`pop()` → returns 3 → Next `pop()` returns 2.