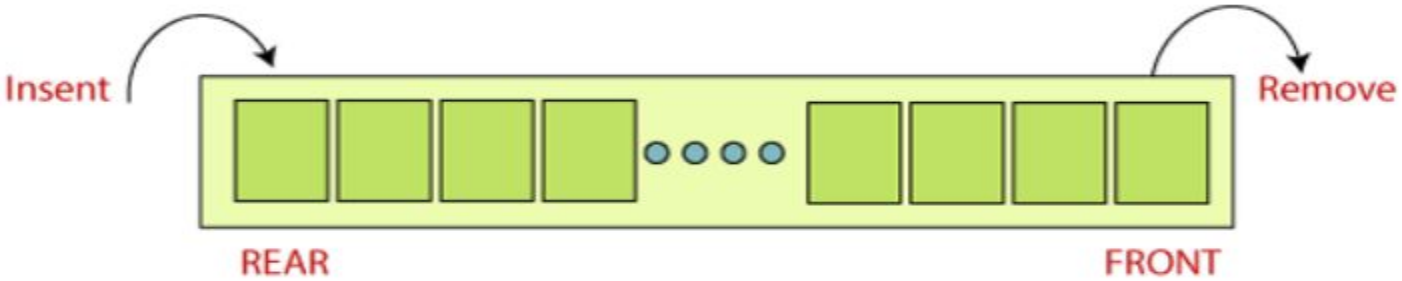
Queues

A Linear Data Structure By:

Bok-Man Victor Siu, Justin Tsuchiyama, Robert Duong, Milan Verma, Damir Rosencrants, Mohamed Abdulla

What is a queue?

- Queue is an abstract data type or a linear data structure, collection designed for holding elements prior to processing
- It is ordered list. Show as following picture, it is FIFO (First in First out - first input into line, then it would be the first one element as output), similar to the ticket queue outside a cinema hall, where the first person entering the queue is the first person who gets the ticket.



Applications of Queue

Use a queue when you want to get things out in the order that you put them in.

Since it is FIFO, queues are good to use as a data structure for line-up processing applications

For example:

API calls, customer waiting line in market

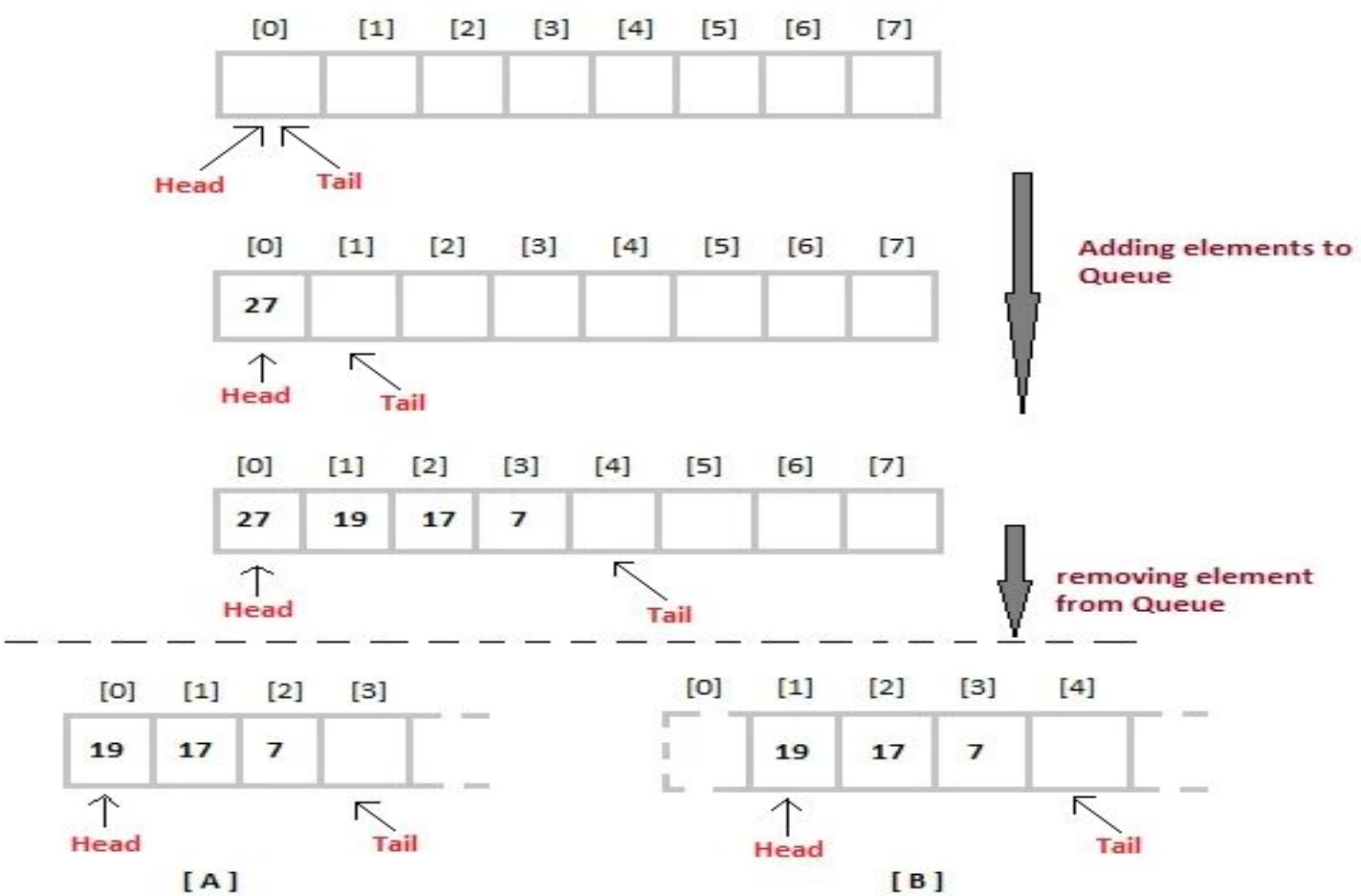
The applications using alert and notification systems, tend to have a FIFO behaviour.

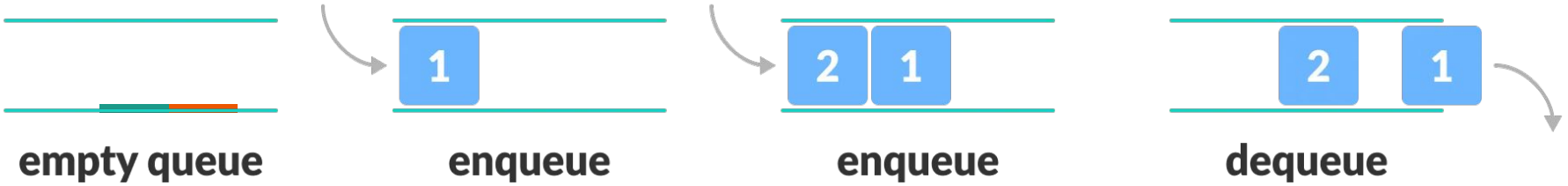
You can think that employees working in call center, when receiving a call first, that is the one is put on hold first.

Ways to implement a Queue

Queue can be implemented using an Array, Stack or Linked List. The easiest way of implementing a queue is by using an Array.

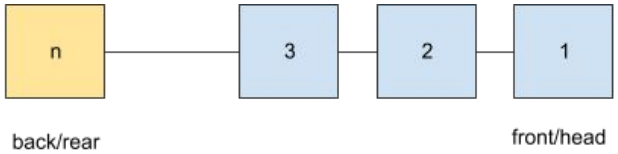
Initially the head(FRONT) and the tail(REAR) of the queue points at the first index of the array (starting the index of array from 0). As we add elements to the queue, the tail keeps on moving ahead, always pointing to the position where the next element will be inserted, while the head remains at the first index.


[0] [1] [2] [3] [4] [5] [6] [7]
Head
Tail
27
Adding elements to Queue
19
17
7
removing element from Queue
[0] [1] [2] [3]
[0] [1] [2] [3] [4]
[A]
[B]

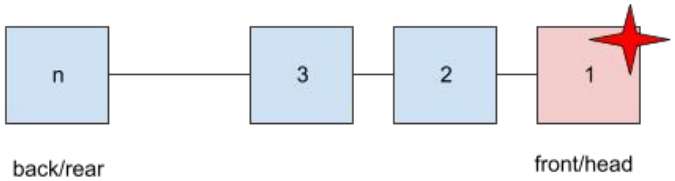

1
2
1
2
1
empty queue
enqueue
enqueue
dequeue

Basic Operations of Queue

Enqueue: Operation inserts an element to the end of the queue.



Dequeue: Operation removes an element from the front of the queue.



Basic Operations of Queue (Cont.)

Size: This operation returns the number of elements in the list.

printQueue: To print the elements of the Queue.

Peek: Get the value from the front of the queue without removing it.

IsEmpty: Check if the queue is empty. Returns a boolean value.

IsFull: Check if the queue is full. Returns a boolean value.

Advantages of Queue

- Queue can handle different data types (Queue of any objects- ints, arrays, sub-queue, strings)
- The array makes the setup of Queue simple (one elements gets added to the end then pushed over 1 every time a new element is added)

Disadvantages of Queue

- Queue is FIFO based (First-In-First-Out), meaning removing elements from the middle is complex
- Queues do not have an index, which makes searching queues tedious.
- Adding new elements into a Queue can only be done by removing prior elements