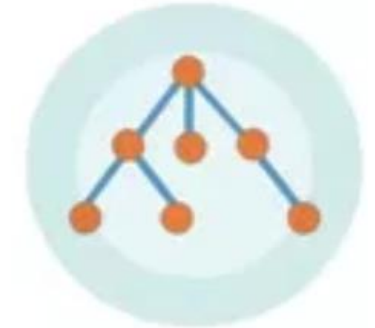
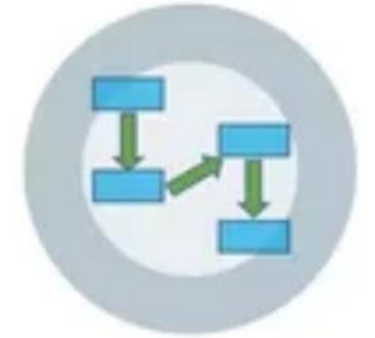


Queue

Circular Queue, Priority Queue, and Deque

Data Structures and Algorithms – **CSI 401**



Arfan Shahzad

{ arfanskp@gmail.com }

Data Structures and Algorithms

Course Contents:

Abstract data types, complexity analysis, Big Oh notation, Stacks (linked lists and array implementations), Recursion and analyzing recursive algorithms, divide and conquer algorithms, Sorting algorithms (selection, insertion, merge, quick, bubble, heap, shell, radix, bucket), queue, dequeuer, priority queues (linked and array implementations of queues), linked list & its various types, sorted linked list, searching an unsorted array, binary search for sorted arrays, hashing and indexing, open addressing and chaining, trees and tree traversals, binary search trees, heaps, M-way tress, balanced trees, graphs, breadth-first and depth-first traversal, topological order, shortest path, adjacency matrix and adjacency list implementations, memory management and garbage collection

Queue

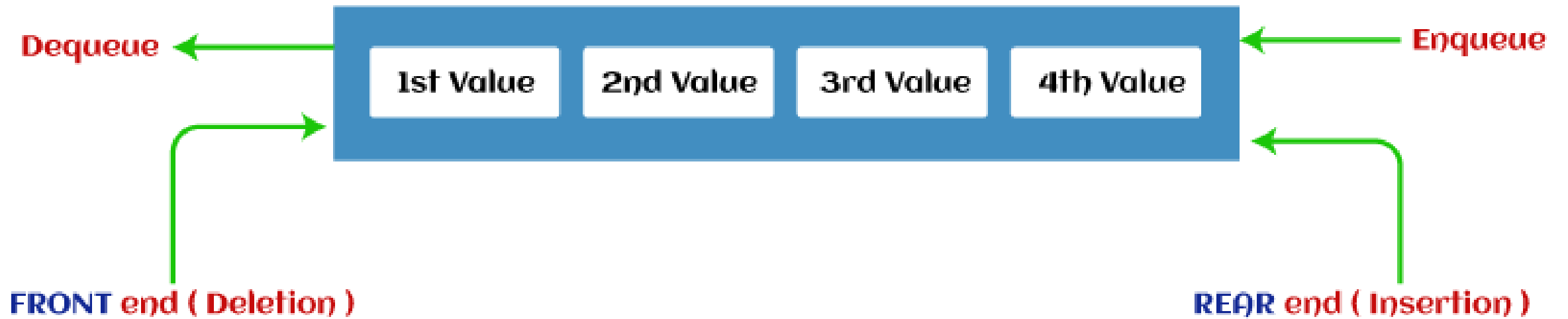
- Formally, the queue abstract data type defines a collection that *keeps objects in a sequence*, where element access and deletion (dequeue) are *restricted to the first element in the sequence*, which is called the front of the queue, and element insertion (enqueue) is *restricted to the end of the sequence*, which is called the rear of the queue.

Queue cont...

- This restriction of insertion and deletion in a queue enforces the rule **first-in first-out (FIFO)**.



Queue cont...



Queue cont...

Methods

- The *queue* abstract data type (ADT) supports the following two fundamental methods:
- **enqueue(*e*)**: Insert element *e* at the rear of the queue.
- **dequeue()**: Remove and return from the queue the object at the front; an error occurs if the queue is empty.

Queue cont...

Methods

- Additionally, the queue ADT includes the following supporting methods:
- **size()**: Return the number of objects in the queue.
- **isEmpty()**: Return a Boolean value that indicates whether the queue is empty.
- **front()**: Return, but do not remove, the front object in the queue; an error occurs if the queue is empty.

Queue cont...

Application of Queues

- Due to the fact that queue performs actions on **first in first out** basis which is quite fair for the ordering.
- There are various applications of queues discussed as below:
 1. Queues are widely used as waiting lists for a single shared resource like printer, disk, CPU.

Queue cont...

Application of Queues

2. Queues are used in asynchronous transfer of data (where data is not being transferred at the same rate between two processes) for e.g. pipes, file IO, sockets.
3. Queues are used as buffers in most of the applications like MP3 media player, CD player, etc.

Queue cont...

Application of Queues

4. Queue are used to maintain the play list in media players in order to add and remove the songs from the play-list.
5. Queues are used in operating systems for handling interrupts.

Queue cont...

Complexity

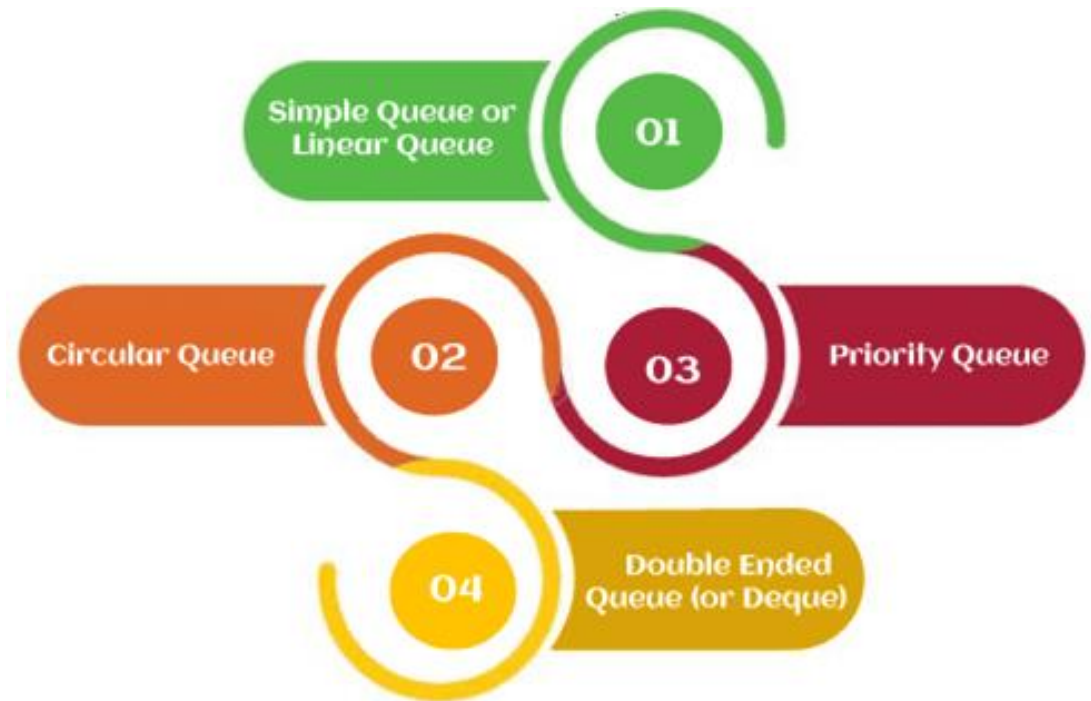
Data Structure	Time Complexity								Space Complexity
	<u>Average</u>				<u>Worst</u>				
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
Queue	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$

Queue cont...

Types of Queue

- There are four different types of queue that are listed as follows:

1. Simple Queue or Linear Queue
2. Circular Queue
3. Priority Queue
4. Double Ended Queue (or Deque)



Queue cont...

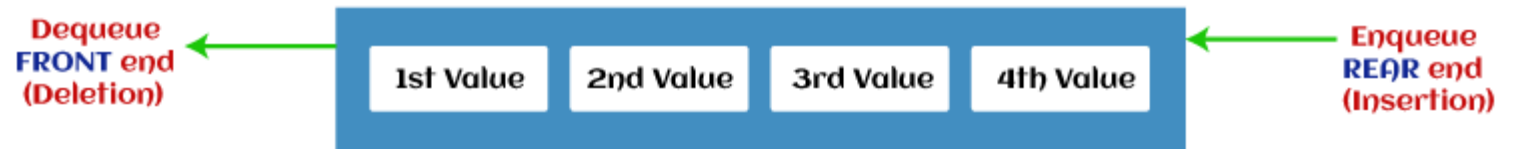
Types of Queue (Simple/ Linear)

- In Linear Queue, an insertion takes place from one end while the deletion occurs from another end.
- The end at which the insertion takes place is known as the rear end, and the end at which the deletion takes place is known as front end.
- It strictly follows the FIFO rule.

Queue cont...

Types of Queue (Simple/ Linear)

- The major drawback of using a linear Queue is that ***insertion is done only from the rear end.***
- In this case, the linear Queue shows the overflow condition as the rear is pointing to the last element of the Queue.



Queue cont...

Types of Queue (Simple/ Linear)

- One another limitation in the array implementation of Queue:
- If the rear reaches to the end position of the Queue then there might be possibility that some vacant spaces are left in the beginning which cannot be utilized.
- So, to overcome such limitations, the concept of the circular queue was introduced.

Queue cont...

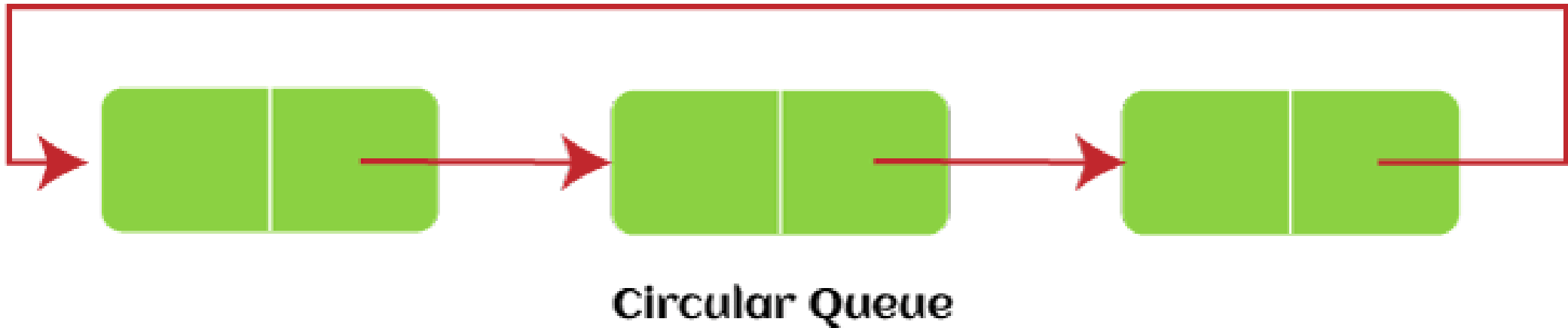
Types of Queue (Circular)

- In Circular Queue, all the nodes are represented as circular.
- It is similar to the linear Queue except that the last element of the queue is connected to the first element.
- It is also known as ***Ring Buffer***, as all the ends are connected to another end.

Queue cont...

Types of Queue (Circular)

- The representation of circular queue is shown in the below image:



Queue cont...

Types of Queue (Circular – Enqueue)

- The steps of enqueue operation are given below:
 1. First, we will check whether the Queue is full or not.
 2. Initially the front and rear are set to -1.
 3. When we insert the first element in a Queue, front and rear both are set to 0.
 4. When we insert a new element, the rear gets incremented, i.e., $\text{rear}=\text{rear}+1$.

Queue cont...

Types of Queue (Circular – Enqueue)

- There are two scenarios in which queue is not full:
- If $\text{rear} \neq \text{max} - 1$, then rear will be incremented to mod (maxsize) and the new value will be inserted at the rear end of the queue.
- If $\text{front} \neq 0$ and $\text{rear} = \text{max} - 1$, it means that queue is not full, then set the value of rear to 0 and insert the new element there.

Queue cont...

Types of Queue (Circular – Enqueue)

- There are two cases in which the element cannot be inserted:
- When $front == 0$ && $rear = max-1$, which means that front is at the first position of the Queue and rear is at the last position of the Queue.
- $front == rear + 1$;

Queue cont...

Types of Queue (Circular – Dequeue)

- The steps of dequeue operation are given below:
 1. First, we check whether the Queue is empty or not. If the queue is empty, we cannot perform the dequeue operation.
 2. When the element is deleted, the value of front gets decremented by 1.
 3. If there is only one element left which is to be deleted, then the front and rear are reset to -1.

Queue cont...

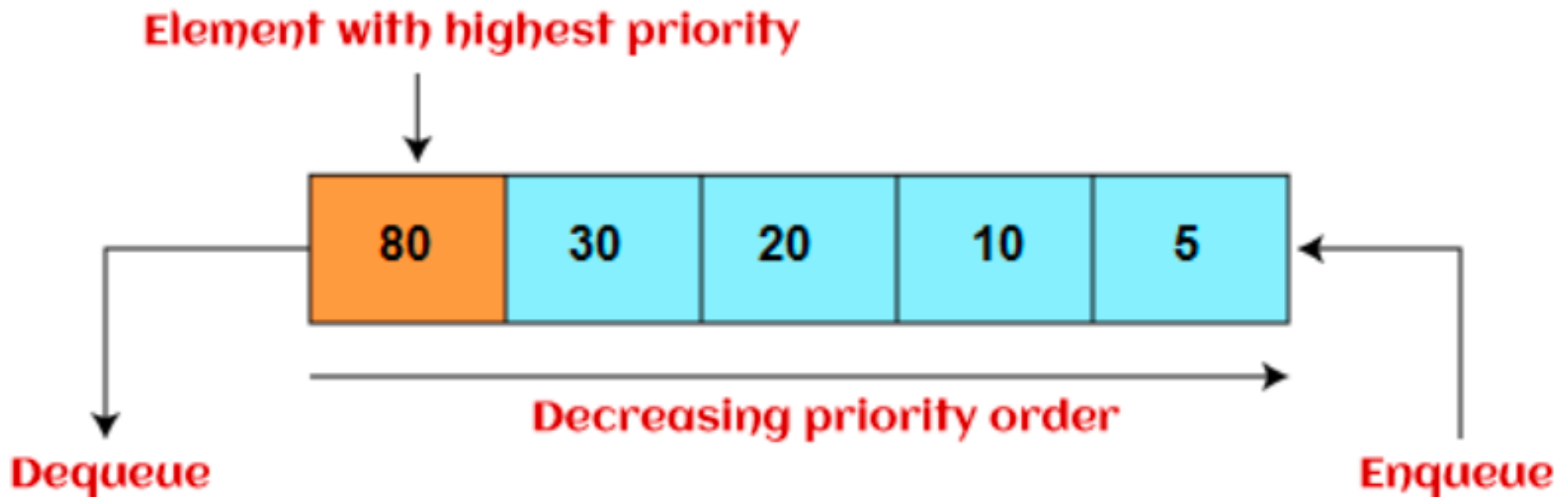
Types of Queue (Priority queue)

- It is a special type of queue in which the elements are *arranged* based on the priority.
- In this data structure every element has a priority associated with it.
- Suppose some elements occur with the *same priority*, they will be arranged according to the FIFO principle.

Queue cont...

Types of Queue (Priority queue)

- The representation of priority queue is shown in the below image:



Queue cont...

Types of Queue (Priority queue)

- Insertion (enqueue) in priority queue takes place based on the arrival, while deletion (dequeue) in the priority queue occurs based on the priority.
- Priority queue is mainly used to implement the CPU scheduling algorithms.
- There are two types of priority queue that are discussed as follows:

Queue cont...

Types of Queue (Priority queue)

- **Ascending priority queue**: In ascending priority queue, elements can be inserted in arbitrary order, but only smallest can be deleted first.
- Suppose an array with elements 7, 5, and 3 in the same order, so, insertion can be done with the same sequence, but the order of deleting the elements is 3, 5, 7.

Queue cont...

Types of Queue (Priority queue)

- **Descending priority queue**: In descending priority queue, elements can be inserted in arbitrary order, but only the largest element can be deleted first. Suppose an array with elements 7, 3, and 5 in the same order, so, insertion can be done with the same sequence, but the order of deleting the elements is 7, 5, 3.

Queue cont...

Types of Queue (Deque)

- A **double-ended queue** (**deque**) is an ADT that generalizes a queue, for which *elements can be added to or removed from either the front (head) or rear (tail)*.
- **Note:** As a verb **dequeue** is to remove an item from a queue.
- The fundamental methods of the **deque** ADT are as follows:

Queue cont...

Types of Queue (Deque)

- **addFirst():** Insert a new element at the beginning of the deque.
- **addLast():** Insert a new element at the end.
- **removeFirst():** Remove and return the first element of the deque.
- **removeLast():** Remove and return the last element of the deque.

Queue cont...

Types of Queue (Deque)

- Additionally, the deque ADT also include the following methods:
- **getFirst()**: Return the first element of deque; an error occurs if deque is empty.
- **getLast()**: Return the last element of deque; an error occurs if deque is empty.
- **size()**: Return the number of elements of the deque.
- **isEmpty()**: Determine if the deque is empty.

Queue cont...

Types of Queue (Deque)

- Deque can be used both as stack and queue as it allows the insertion and deletion operations on both ends.
- Deque can be considered as stack because stack follows the LIFO (Last In First Out) principle in which insertion and deletion both can be performed only from one end.
- Furthermore in deque, it is possible to perform both insertion and deletion from one end, and Deque does not follow the FIFO principle.

Queue cont...

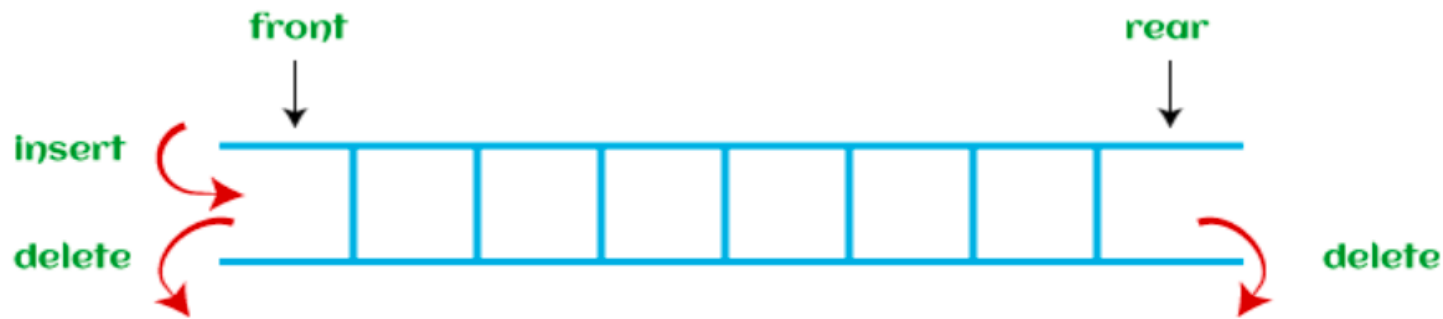
Types of Queue (Deque)

- Deque can be used both as stack and queue as it allows the insertion and deletion operations on both ends.
- Deque can be considered as stack because stack follows the LIFO (Last In First Out) principle in which insertion and deletion both can be performed only from one end.
- Furthermore in deque, it is possible to perform both insertion and deletion from one end, and Deque does not follow the FIFO principle.

Queue cont...

Types of Queue (Deque)

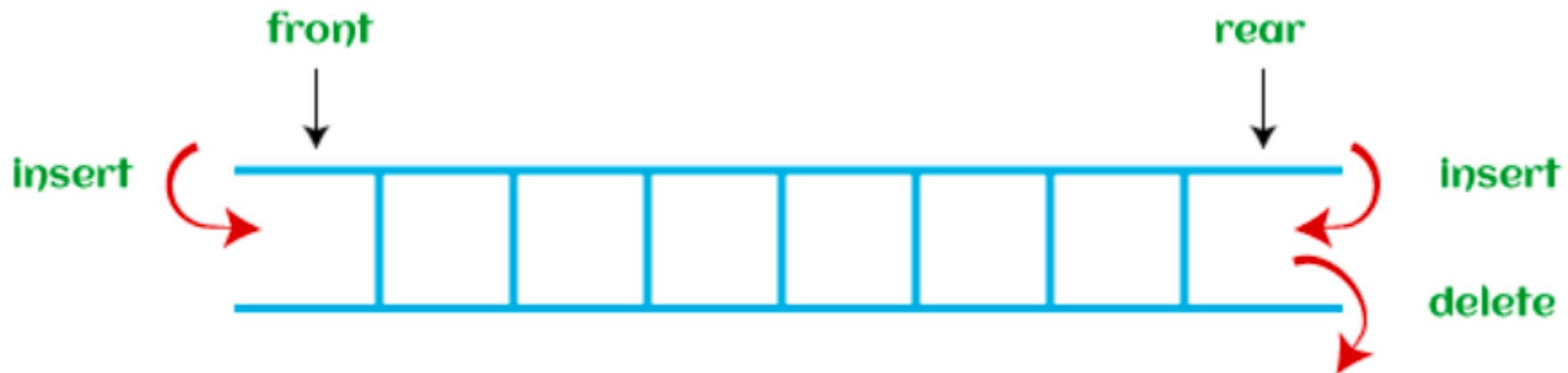
- There are two types of deque that are discussed as follows:
- **Input restricted deque**: As the name implies, in input restricted queue, insertion operation can be performed at only one end, while deletion can be performed from both ends.



Queue cont...

Types of Queue (Deque)

- **Output restricted deque**: As the name implies, in output restricted queue, deletion operation can be performed at only one end, while insertion can be performed from both ends.



Queue cont...

Implementation of Queues

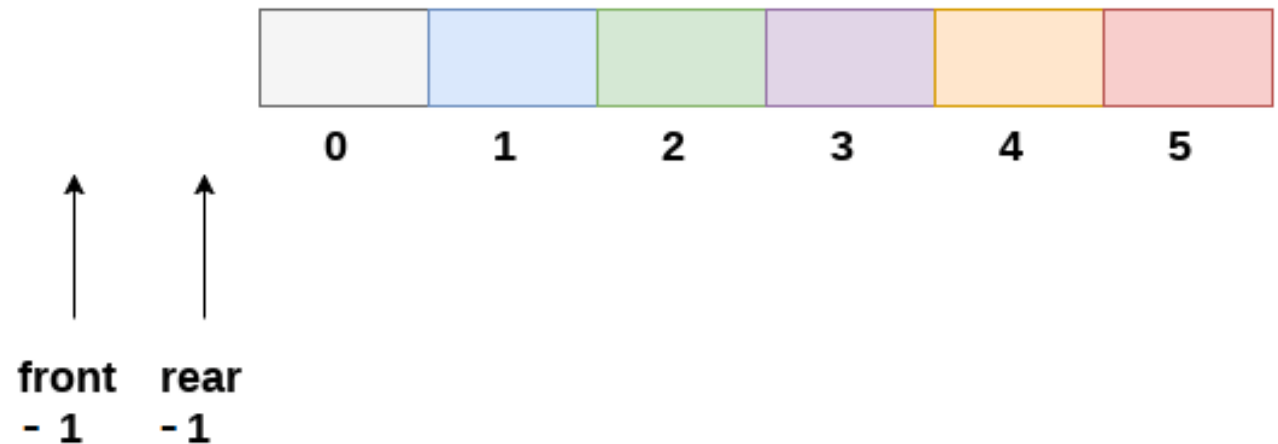
- There are two ways of implementing the Queue:
 1. **Implementation using array**: The sequential allocation in a Queue can be implemented using an array.
 2. **Implementation using Linked list**: The linked list allocation in a Queue can be implemented using a linked list.

Queue implementation using Array

- We can easily implement queue by using linear arrays.
- There are two variables i.e. front and rear, that are implemented in the case of every queue.
- Front and rear variables point to the position from where insertions and deletions are performed in a queue.

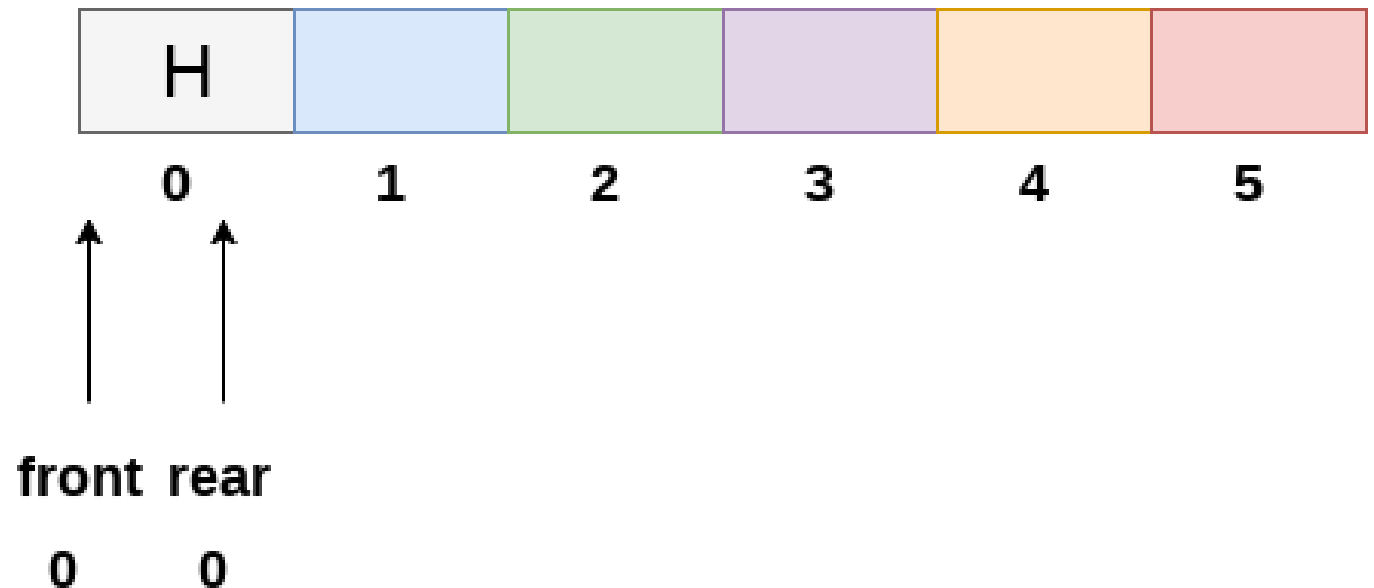
Queue implementation using Array cont...

- Initially, the value of front and rear of queue is -1 which represents an empty queue.



Queue implementation using Array cont...

- When we insert the first element in a Queue, **front** and **rear** both are set to 0.

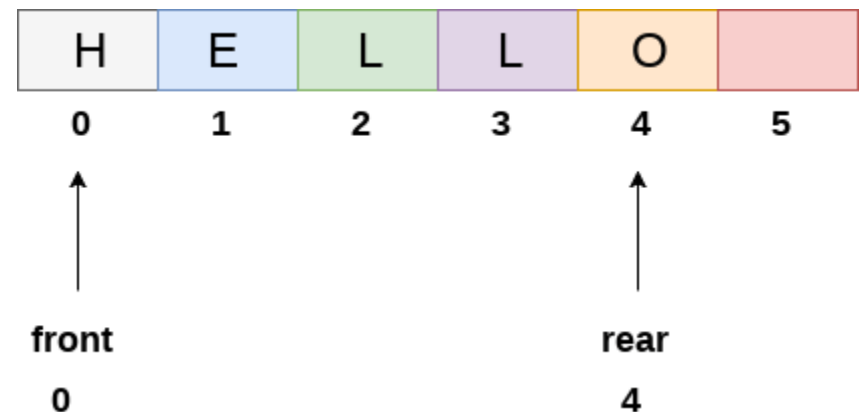


Queue implementation using Array cont...

- Whenever we insert a new element, the rear gets incremented, i.e., $\text{rear}=\text{rear}+1$.
- When we delete an element, the front gets incremented, i.e., $\text{front}=\text{front}+1$.
- Array implementation of a queue containing 5 elements along with the respective values of front and rear, is shown in the following figure.

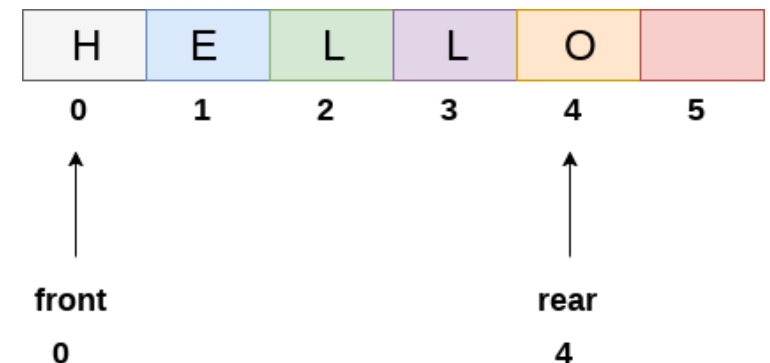
Queue implementation using Array cont...

- The following figure shows the queue of characters forming the English word "HELLO".
- However, the value of rear increases by one every time an insertion is performed in the queue.

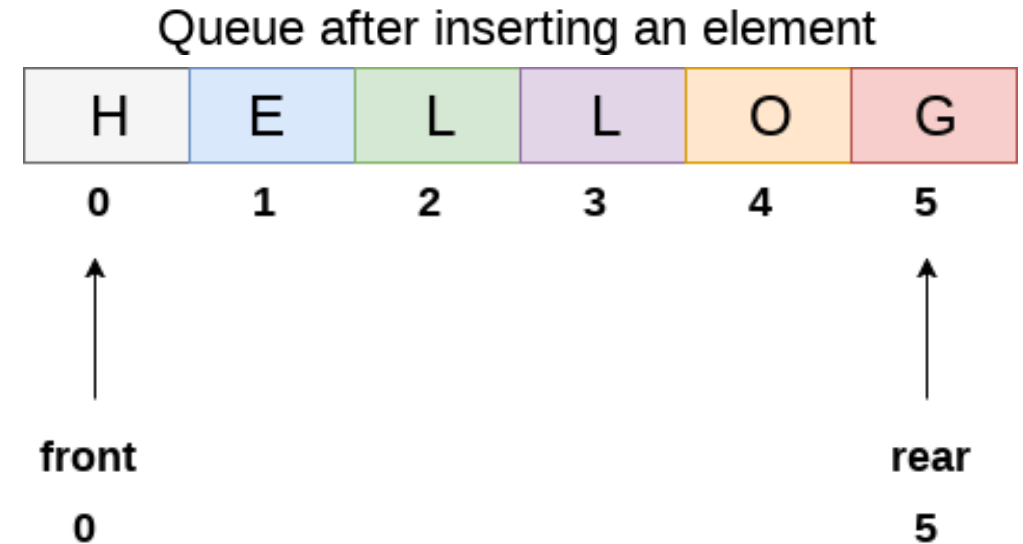


Queue implementation using Array cont...

- Since, No deletion is performed in the queue till now, therefore the value of front remains 0.
- After inserting an element into the queue shown in the figure, the queue will look something like following:



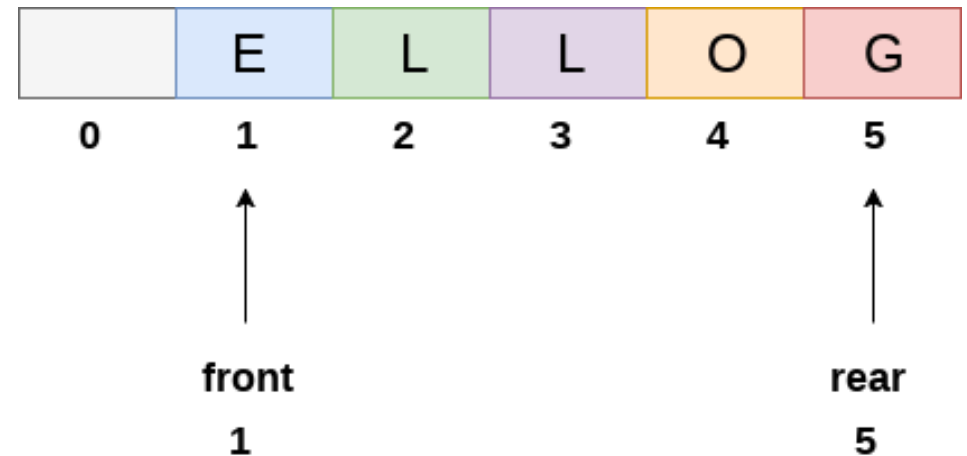
Queue implementation using Array cont...



- The value of rear will become 5 while the value of front remains same.

Queue implementation using Array cont...

- After deleting an element, the value of front will increase from -1 to 0. however, the queue will look something like following.



Queue after deleting an element

Queue implementation using Array cont...

Algorithm to insert any element in a queue

- Check if the queue is already full by comparing rear to max - 1.
- if so, then return an overflow error.
- If the item is to be inserted as the first element in the list, in that case set the value of front and rear to **0** and insert the element at the **rear end**.

Queue implementation using Array cont...

Algorithm to insert any element in a queue

- Otherwise keep increasing the value of rear and insert each element one by one having rear as the index.

Queue implementation using Array cont...

Algorithm to insert any element in a queue

- Step 1: IF REAR = MAX - 1
- Write OVERFLOW
- Go to step
- [END OF IF]

Queue implementation using Array cont...

Algorithm to insert any element in a queue

- **Step 2:** IF FRONT = -1 and REAR = -1
- SET FRONT = REAR = 0
- ELSE
- SET REAR = REAR + 1
- [END OF IF]

Queue implementation using Array cont...

Algorithm to insert any element in a queue

- Step 3: Set $QUEUE[REAR] = NUM$
- Step 4: EXIT

Queue implementation using Array cont...

Algorithm to delete any element in a queue

- If, the value of front is **-1** or value of front is ***greater than rear***, write an ***underflow message*** and **exit**.
- Otherwise, keep ***increasing the value*** of front and ***return the item*** stored at the front end of the queue at each time.

Queue implementation using Array cont...

Algorithm to delete any element in a queue

- **Step 1:** IF FRONT = -1 or FRONT > REAR
- Write UNDERFLOW
- ELSE
- SET VAL = QUEUE[FRONT]
- SET FRONT = FRONT + 1
- [END OF IF]
- **Step 2:** EXIT