

# Queue Data Structure & Applications

---

**SUNBEAM**  
Data Structure  
Trainer : Smita Kadam  
Email ID : [smita@sunbeaminfo.com](mailto:smita@sunbeaminfo.com)



# Queue Data Structure

- Collection of similar type of elements. Element can be always added at rear position and element can be deleted always from front position.
- Follows FIFO. Element which joins queue first gets chance to leave queue first.
- **Queue can be implemented**
  - **Using Static Implementation of Memory**
    - Queue size can not be increased or decreased at runtime.
    - Can be used when queue size is predefined
  - **Using Dynamic Implementation of Memory**
    - Queue size can be increased or decreased at runtime.
    - Optimized usage of memory



# Queue Data Structure

- Operations can be performed on queue
  - **Enqueue/Add/Join/Insert** – adds a element on queue
    - isfull - checks full state of queue

*rear*  
 $top == SIZE-1 ? TRUE : FALSE$
  - **Dequeue/Delete/Leave/remove** – deletes element from queue
    - isempty – checks empty state of queue

*front*  
 $top == -1 ? TRUE : FALSE$   
 $f > R$
  - **Peek** – retrieves element from queue
    - isempty – checks empty state of stack

*front*  
 $top == -1 ? TRUE : FALSE$   
 $f > R$
  - **Poll** – retrieves element and delete from queue. This operation is available with priority queue.
    - isempty – checks empty state of stack

*f*  
 $top == -1 ? TRUE : FALSE$



# Variations of Queue

- Variations of queue are available in four types. Each variation manage front and rear as per its own type.
  - Linear Queue
  - Circular Queue
  - Priority Queue
  - DeQueue
    - **Input Restricted Queue** : is a dequeue which allows insertion at only one end i.e. rear end, but allows deletion at both ends, rear and front end of the lists.
    - **Output Restricted Queue** : is a queue, which allows deletion at only one end i.e. front end, but allows insertion at both ends rear and front end of the list.



# Applications of Queue

- **Linear Queue**

- Managing requests on a single shared resource such as CPU scheduling and disk scheduling.
- Handling hardware or real-time systems interrupts.
- Handling website traffic.
- Routers and switches in networking.
- Maintaining the playlist in media players.

- **Circular Queue**

- Memory Management
- Process Scheduling
- Traffic Systems



# Applications of Queue

- **Priority Queue**

- It is used in prim's algorithm.
- It is used in the Dijkstra's shortest path algorithm.
- It is used in data compression techniques like Huffman code.
- It is used in heap sort.
- It is also used in operating system like priority scheduling, load balancing and interrupt handling.

- **DEQueue Queue**

- The A-steal algorithm implements task scheduling for multiple processors (multiprocessor scheduling).

The processor gets the first element from the double ended queue. - When one of the processors completes execution of its own thread, it can steal a thread from other processors.





Thank you!

