Queues

Definition:

A queue is a fundamental data structure that follows the First In, First Out (FIFO) principle. This means that the first element added to the queue is the first one to be removed.

Example:

The first person who joins the queue is the first one to be served.

Characteristics of Queues:

1. **FIFO Principle:**
    - The First In, First Out principle dictates the order in which elements are processed in a queue.
2. **Operations:**
    - **Enqueue (or Push):** Adds an element to the back (or rear) of the queue.
    - **Dequeue (or Pop):** Removes the element from the front (or front) of the queue.
    - **Front (or Peek):** Retrieves the element at the front of the queue without removing it.
3. **Implementation:**

- Queues can be implemented using arrays or linked lists. Arrays are efficient when the size of the queue is fixed, while linked lists are more flexible in handling dynamic sizes.

Applications of Queues:

1. **Job Scheduling:**
   - Queues are used in job scheduling algorithms, ensuring that tasks are processed in the order they arrive.
2. **Print Queue:**
   - Print jobs in a printer queue are processed in the order they are received.
3. **Breadth-First Search (BFS) in Graphs:**
   - Queues are essential in BFS traversal of graphs, exploring nodes level by level.
4. **Task Management in Operating Systems:**
   - Queues are used to manage tasks in operating systems, such as process scheduling.
5. **Buffer Management:**
   - Queues are employed in managing buffers, ensuring data is processed in the order it is received.

Time Complexity:

1. **Enqueue, Dequeue, and Front:**
   - In most cases, the time complexity for these operations is O(1), meaning they have constant time complexity.
2. **Space Complexity:**

- The space complexity of a queue is O(n), where n is the number of elements in the queue.

Write about Each operation clearly with an example and code

Difference Between Stack and Queue:

Feature	Stack	Queue
Order Principle	Last In, First Out (LIFO)	First In, First Out (FIFO)
Insertion Operation	Push (adds to the top)	Enqueue (adds to the back)
Removal Operation	Pop (removes from the top)	Dequeue (removes from the front)
Access Point	Only the top element is accessible	Front and rear elements are accessible
Real-life Analogy	Stack of plates	A line of people waiting in a queue
Applications	Function call management, expression evaluation, undo mechanisms	Job scheduling, print queues, breadth-first search in graphs
Time Complexity	Push, Pop, and Peek: O(1)	Enqueue, Dequeue, and Front: O(1)
Space Complexity	O(n), where n is the number of elements	O(n), where n is the number of elements

Difference Between Enqueue and Dequeue:

Feature	Enqueue	Dequeue
Definition	Adds an element to the back	Removes an element from the front
Characteristic	Follows the FIFO principle	Follows the FIFO principle
Purpose	Inserts an element at the end	Removes the next element to be processed
Example	Adding a print job to the print queue	Removing the front print job from the print queue
Implementation	Adds an element to the end of the data structure	Removes an element from the front of the data structure