**LAR Liyanage**

**27336**

**Tutorial 03**

1. **What is a queue?**

* A queue is a linear data structure in computer science that follows the First In First Out (FIFO) principle. It is a collection of elements in which elements are added at the end (rear) and removed from the front. In other words, the element that has been in the queue the longest is the first one to be removed.

1. **Draw the logical representation of the Queue?**

**front <---------------------------------- rear**

**^ ^**

**| |**

**+---+ +---+ +---+ +---+ +---+ |**

**| A | -> | B | -> | C | -> | D | -> | E | -> |**

**+---+ +---+ +---+ +---+ +---+ |**

1. **What are the features of a queue?**

* It is a data structure that stores a collection of elements.
* It follows the First-In-First-Out (FIFO) principle, which means that the first element added to the queue will be the first one to be removed.
* It has two main operations:
* Enqueue. (add an element to the back of the queue)
* Dequeue. (remove the element at the front of the queue)
* It allows access to only two elements: the front (or head) of the queue, which is the element that will be dequeued next, and the rear (or tail) of the queue, which is the last element that was added.
* It can be implemented using an array or a linked list data structure.

1. **What is FIFO? Describe.**

* FIFO stands for First-In-First-Out, and it is a principle that is used in queue data structures. It means that the first element added to the queue will be the first one to be removed. In other words, the element that has been in the queue the longest will be the first one to be dequeued. This is in contrast to the Last-In-First-Out (LIFO) principle, which is used in stack data structures, where the last element added is the first one to be removed. The FIFO principle is commonly used in situations where the order in which elements are added to the data structure is important, such as in job scheduling or network packet processing.

1. **There are three instances in which the queue could be empty. Mention those?**

* Initial State :- When the queue is first created and no elements have been added yet, it will be empty.
* Dequeuing all elements :- If all the elements in the queue have been dequeued or removed, the queue will become empty.
* Dequeue operation on an empty queue :- If you attempt to dequeue an element from an already empty queue, the queue will remain empty.

1. **Give examples of queues in real life.**

* Waiting in line :- Standing in a queue at a ticket counter, a bank, or a supermarket is a real-life example of a queue. The person who enters first gets serviced first, and new people join the queue at the rear.
* Print spooler :- When multiple users send print jobs to a printer, they are placed in a queue. The printer processes the print jobs in the order they entered the queue, ensuring fairness.
* Online ticket booking :- When several people attempt to book tickets simultaneously for a popular event or movie, their requests are placed in a queue. The requests are processed one by one, giving each person a chance to book.
* Customer support hotline :- When customers call a customer support hotline, their calls are placed in a queue. The support agents attend to each call in the order it was received, ensuring fairness.
* Message passing systems :- In distributed systems, message queues are used to pass messages between various components or processes. The messages are sequentially processed based on their order in the queue.