Computer Networking Systems Department

Data structure

DANIAH ABDUL QAHAR SHAKIR

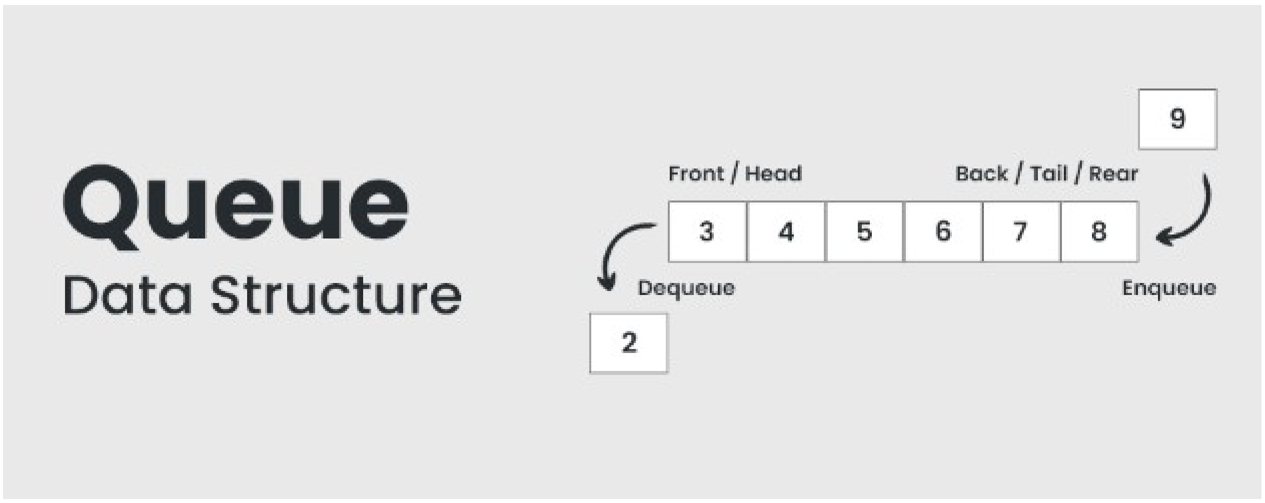
2024-2025

Lecture 4

First course

A **Queue** is defined as a linear data structure that is open at both ends and the operations are performed in First In First Out (FIFO) order.

We define a queue to be a list in which all additions to the list are made at one end, and all deletions from the list are made at the other end. The element which is first pushed into the order, the operation is first performed on that.



FIFO Principle of Queue:

- A Queue is like a line waiting to purchase tickets, where the first person in line is the first person served. (i.e. First come first serve).
- Position of the entry in a queue ready to be served, that is, the first entry that will be removed from the queue, is called the **front** of the queue(sometimes, **head** of the queue), similarly, the position of the last entry in the queue, that is, the one most recently added, is called the **rear** (or the **tail**) of the queue. See the below figure.

What is better, a stack or a queue?

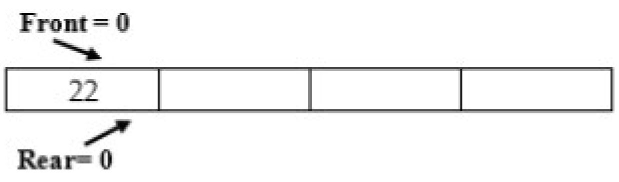
If you want things to come out in the order you put them in, use a queue. Stacks are useful when you want to reorder things after putting them in.

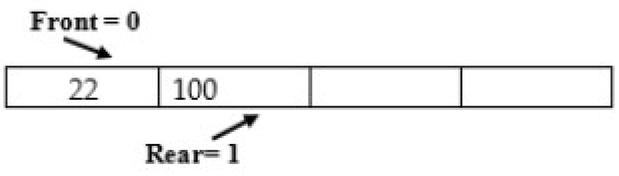
Primary Queue Operations:

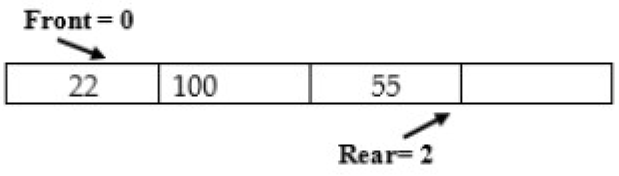
- ✓ **void enqueue(int Element):** When this operation is performed, an element is inserted in the queue at the end i.e. at the rear end. (Where T is Generic i.e we can define Queue of any type of data structure.)
- ✓
- ✓ **int dequeue():** When this operation is performed, an element is removed from the front end and is returned.

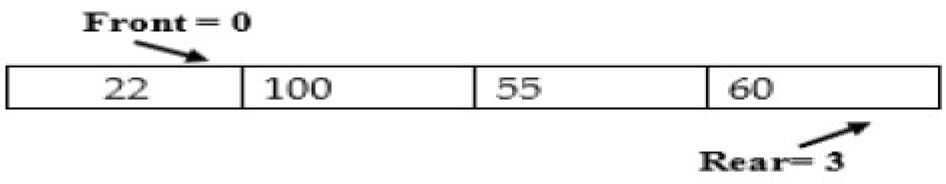
Initial queue
Front = -1
Rear= -1

Enqueuing operations

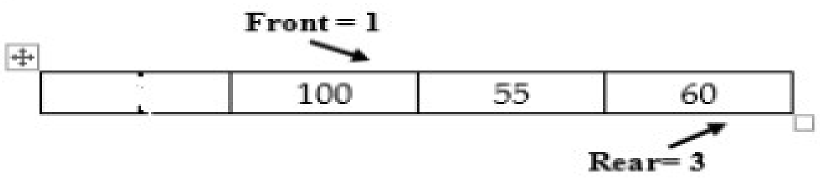


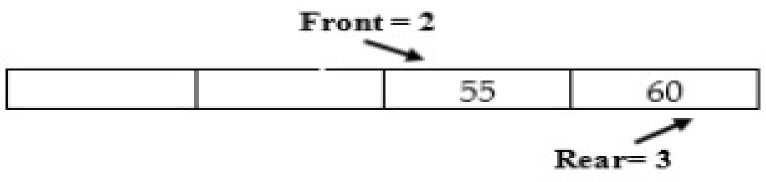


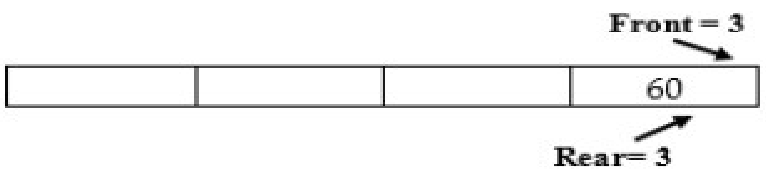




Dequeuing operations







THERE ARE TWO ALGORITHMS TO INSERT (ENQUEUING) AND REMOVE (DEQUEUING) ITEMS INTO AND FROM QUEUE.

❖ **Insert Algorithm (Enqueuing)**

```
•[overflow]
            if R>=N-1
                  Then Over flow
•[increment Rear]

            Rear  =  Rear+1
•[insert element]

            Queue[Rear]  =  New element
•[set Front]
              ▪ If Front= -1
                Then Front =0
```

❖ **Delete or retrieve Algorithm (Dequeuing)**

```
•[underflow]
            if front = -1
                  Then underflow
• Delete or retrieve element
         Element = Queue [ Front]

•[Check empty queue ]
          If Front =Rear
            Then Front =Rear= -1
 Else Front=Front+1
```

✓ **Types of Queues:**

- **Simple Queue:** Simple queue also known as a linear queue is the most basic version of a queue. Here, insertion of an element i.e. the Enqueue operation takes place at the rear end and removal of an element .

- **Circular Queue:** In a circular queue, the element of the queue act as a circular ring. The working of a circular queue is similar to the linear queue except for the fact that the last element is connected to the first element.

- **Priority Queue:** This queue is a special type of queue. Its specialty is that it arranges the elements in a queue based on some priority.

- **Dequeue:** Dequeue is also known as Double Ended Queue. As the name suggests double ended, it means that an element can be inserted or removed from both the ends of the queue unlike the other queues in which it can be done only from one end. Because of this property it may not obey the First In First Out property

.

Implementation of Queue:

- **Sequential allocation:** A queue can be implemented using an array. It can organize a limited number of elements.
- **Linked list allocation:** A queue can be implemented using a linked list. It can organize an unlimited number of elements.

Applications of Queue:

- **Multi programming:** Multi programming means when multiple programs are running in the main memory. It is essential to organize these multiple programs and these multiple programs are organized as queues.
- **Network:** In a network, a queue is used in devices such as a router or a switch. another application of a queue is a mail queue which is a directory that stores data and controls files for mail messages.
- **Job Scheduling:** The computer has a task to execute a particular number of jobs that are scheduled to be executed one after another. These jobs are assigned to the processor one by one which is organized using a queue.
- **Shared resources:** Queues are used as waiting lists for a single shared resource.

Real-time application of Queue:

- ATM Booth Line
- Ticket Counter Line
- Key press sequence on the keyboard
- CPU task scheduling
- Waiting time of each customer at call centers.

Advantages of Queue:

- A large amount of data can be managed efficiently with ease.
- Operations such as insertion and deletion can be performed with ease as it follows the first in first out rule.
- Queues are useful when a particular service is used by multiple consumers.
- Queues are fast in speed for data inter-process communication.
- Queues can be used in the implementation of other data structures.

Disadvantages of Queue:

- The operations such as insertion and deletion of elements from the middle are time consuming.
- Limited Space.
- In a classical queue, a new element can only be inserted when the existing elements are deleted from the queue.
- Searching an element takes O(N) time.
- Maximum size of a queue must be defined prior.