

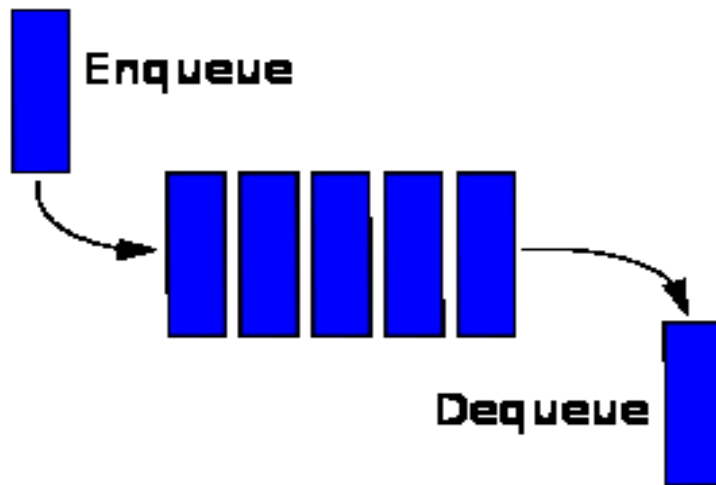
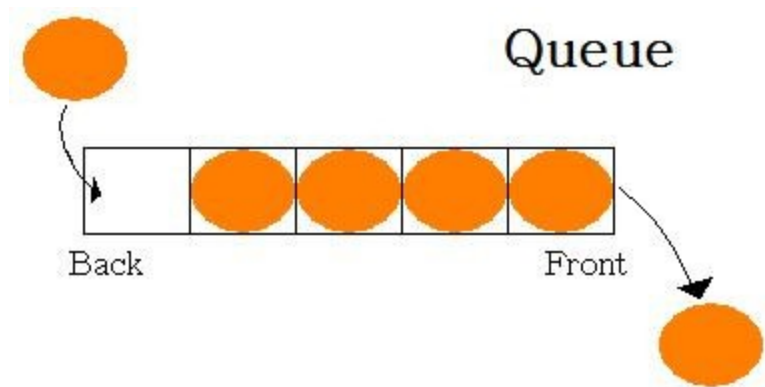
# ***QUEUE***



# Queue

- Ordered collection of homogeneous elements
- Non-primitive linear data structure.
- A new element is added at one end called **rear end** and the existing elements are deleted from the other end called **front end**.
- This mechanism is called First-In-First-Out (**FIFO**).
- Total no of elements in queue =  $\text{rear} - \text{front} + 1$

# Fig: Models of a Queue

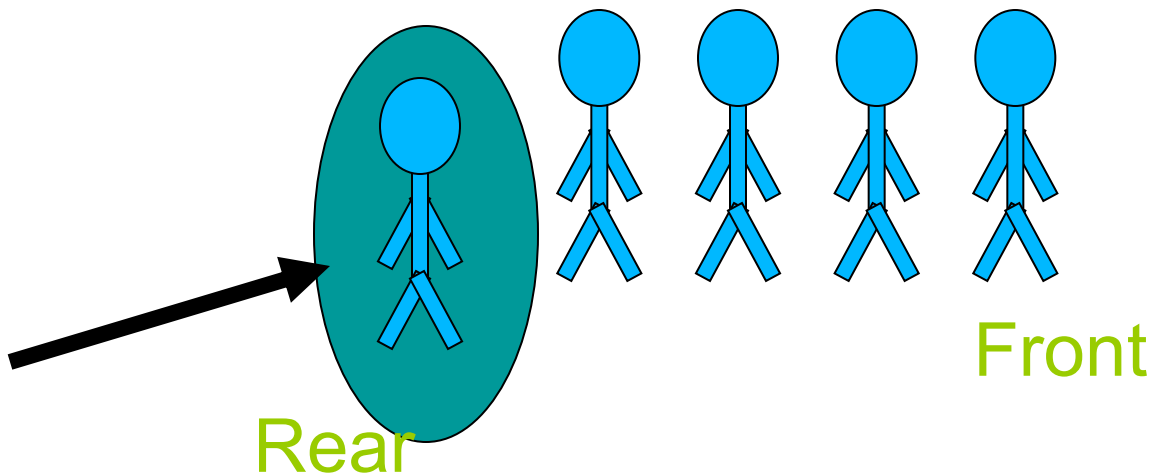


# Operations On A Queue

1. To insert an element in queue
2. Delete an element from queue

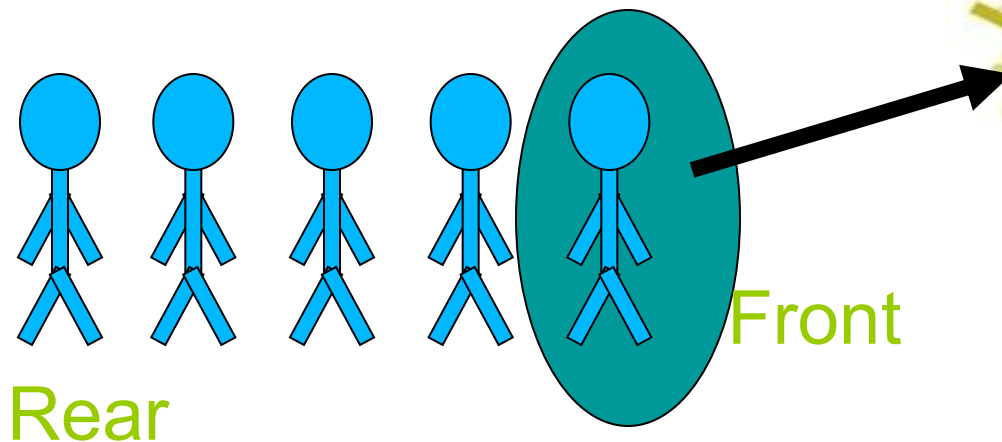
# The Queue Operation

Placing an item in a queue is called “insertion or **enqueue**”, which is done at the end of the queue called “**rear**”.



# The Queue Operation

Removing an item from a queue is called “deletion or **dequeue**”, which is done at the other end of the queue called “**front**”.



# Algorithm QINSERT (ITEM)

1.If (rear = maxsize-1 )

    print (“queue overflow”) and return

2.Else

    rear = rear + 1

    Queue [rear] = item

# Algorithm QDELETE ()

1. If (front = rear)

    print “queue empty” and return

2. Else

    Front = front + 1

    item = queue [front];

    Return item



# Queue Applications

- Real life examples
  - ✓ Waiting in line
  - ✓ Waiting on hold for tech support
- Applications related to Computer Science
  - ✓ Round robin scheduling
  - ✓ Job scheduling (FIFO Scheduling)
  - ✓ Key board buffer

# 3 states of the queue

1. Queue is empty

$$\text{FRONT}=\text{REAR}$$

2. Queue is full

$$\text{REAR}=\text{N}$$

3. Queue contains element  $\geq 1$

$$\text{FRONT}<\text{REAR}$$

$$\text{NO. OF ELEMENT}=\text{REAR}-\text{FRONT}+1$$

# Representation Of Queues

1. Using an array
2. Using linked list

