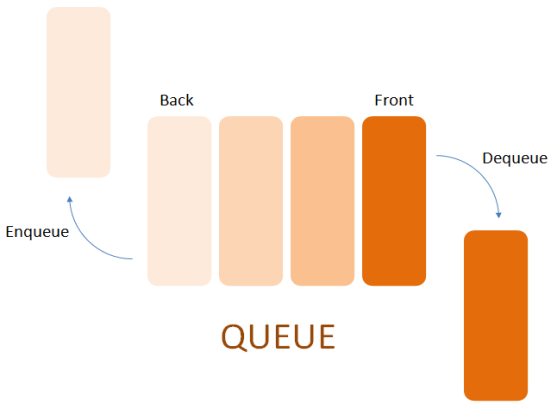
Chapter 9:

Data Structure -2
Queue Using Linear List

By: Jagdish Devrani
Lecturer in Computer Science
Ahlcon Public School

Introduction

In ordinary English, a queue is defined as a waiting line, like a line of people waiting to purchase tickets, where the first person in line is the first person served.



REAL LIFE EXAMPLES OF QUEUE ARE:

1. A queue of people at ticket-window: The person who comes first gets the ticket first. The person who is coming last is getting the tickets in last. Therefore, it follows first-in-first-out (FIFO) strategy of queue.

2. Luggage checking machine: Luggage checking machine checks the luggage first that comes first. Therefore, it follows FIFO principle of queue.

3. Patients waiting outside the doctor's clinic: The patient who comes first visits the doctor first, and the patient who comes last visits the doctor last. Therefore, it follows the first-in-first-out (FIFO) strategy of queue.

IN COMPUTER APLICATION

- For computer applications, we similarly 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.
- Queues are also called first-in, first-out lists , or FIFO for short.

IN COMPUTER APLICATION

As in all parts of life, it is often necessary to wait one's turn before having access to something.

Within a computer system there may be queues of tasks waiting for the printer, for access to disk storage etc…

COMPUTER APPLICATION EXAMPLES

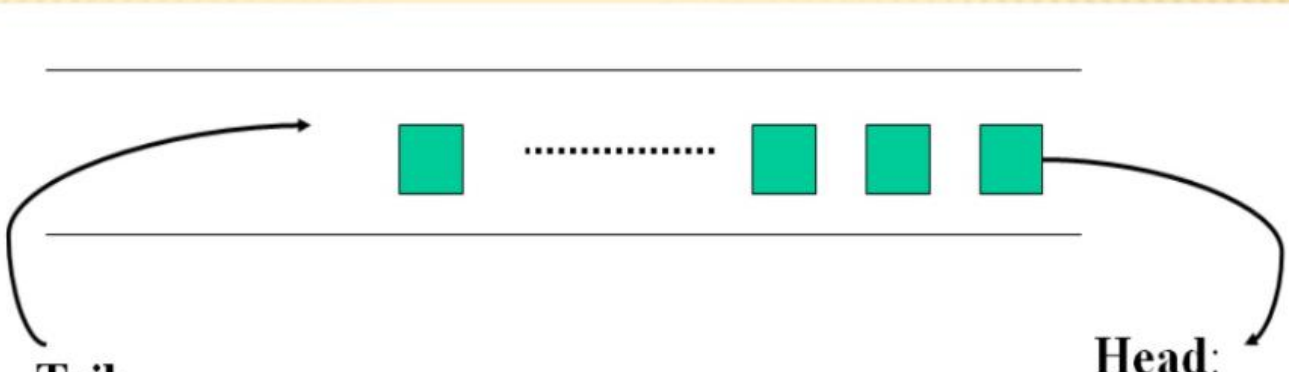
- Imagine you have a web-site which serves files to thousands of users. You cannot service all requests, you can only handle say 100 at once. A fair policy would be first-come-first serve: serve 100 at a time in order of arrival. A Queue would definitely be the most appropriate data structure.
- Similarly in a multitasking operating system, the CPU cannot run all jobs at once, so jobs must be batched up and then scheduled according to some policy. Again, a queue might be a suitable option in this case.
- Say you have a number of documents to be printed at once. Your OS puts all of these docs in a queue and sends them to the printer. The printer takes and prints

Difference between STACK & QUEUE

STACK — ELEMENTS ARE PULLED IN LAST-IN FIRST-OUT-ORDER (E.G. A STACK OF PAPERS)

QUEUE — ELEMENTS ARE PULLED IN FIRST-IN FIRST-OUT-ORDER (E.G. A LINE IN A CAFETERIA)

A Graphic Model of a Queue



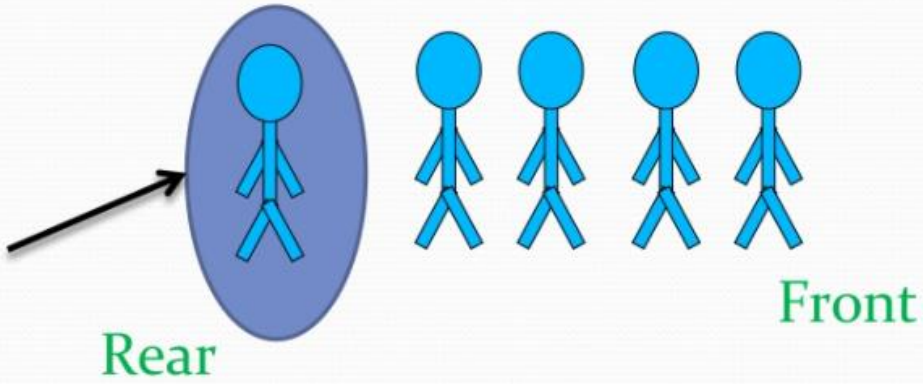
Tail:
All new items are added on this end

Head:
All items are deleted from this end

Operations on Queue

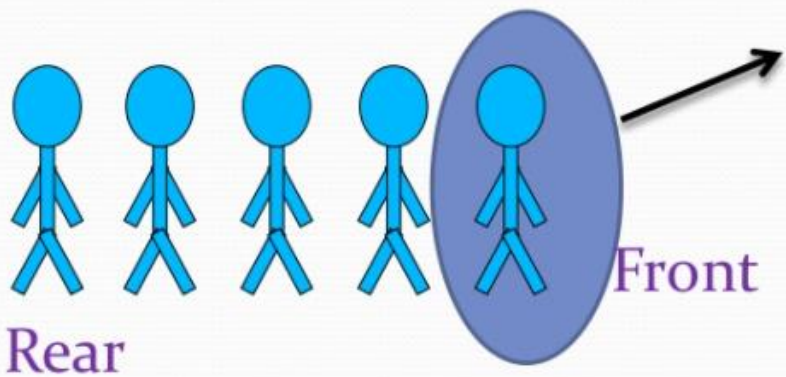
1.Insertion :

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

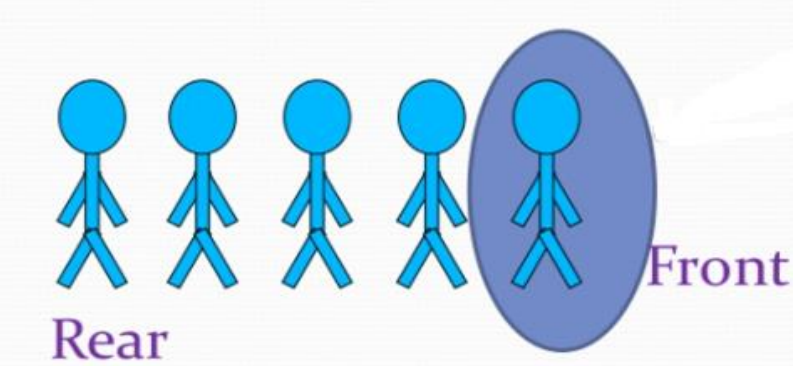


2.Deletion :

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



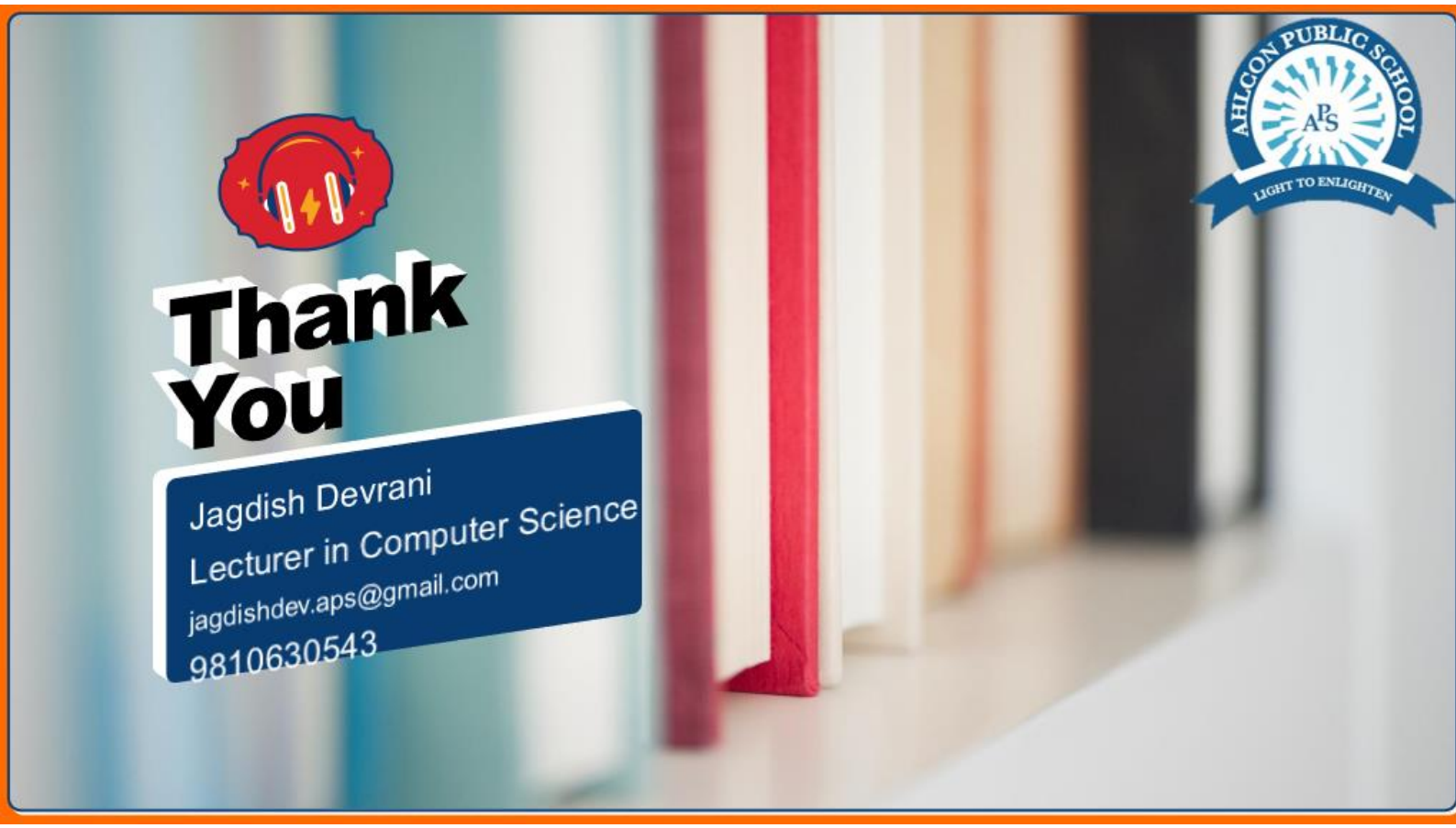
3. Traversing in a Queue




```
#WAP to perform the following operation in a Queue using List
 #          1. Insert      2. Delete   3. Display   4. exit

from collections import deque

def isEmpty(queue):
    if (queue==[]):
        return True
    else:
        return False

def Insert(queue,item):
    global front,rear
    queue.append(item)
    if(front==-1):
        front=0
    rear=rear+1
```

```
def Delete(queue):
    global front,rear
    if isEmpty(queue):
        return "Underflow"
    else:
        item=queue.popleft()
        rear=rear-1
        if(front>rear):
            front=rear=-1
        return item

def Display(queue):
    global front,rear
    print("front :",front," rear :",rear)
    if (isEmpty(queue)):
        print("Underflow Queue is empty can not display")
    else:
        for i in range(front,rear+1):
            print (queue[i], end =" >")
```

```
#_main_
Queue=deque([])
front=rear=-1
ans='y'
while(ans=='y'):
    print("Queue Operations")
    print("Press 1 to Insert")
    print("Press 2 to Delete")
    print("Press 3 to Display")
    print("Press 4 to exit")
    ch=int(input("Enter your choice"))
    if (ch==1):
        item=int(input("Enter Item Value"))
        Insert(Queue,item)
    elif(ch==2):
        item=Delete(Queue)
        if (item=="Underflow"):
            print("Queue is Empty")
        else:
            print("Deleted item is ",item)
    elif(ch==3):
        Display(Queue)
    elif(ch==4):
        break
    else:
        print("Invalid choice")
    ans=input("Do you wish to continue")
```


Thank
You
Jagdish Devrani
Lecturer in Computer Science
jagdishdev.aps@gmail.com
9810630543
AHLCON PUBLIC SCHOOL
APS
LIGHT TO ENLIGHTEN