N. Selvakumar
AP/CSE.

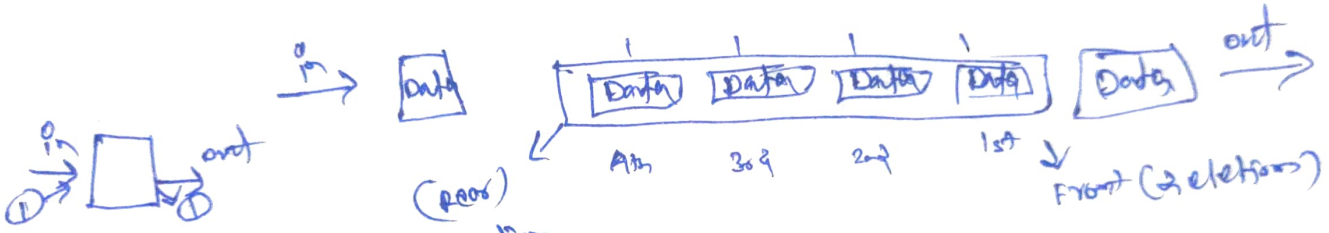
Queues ADT.

Definition:

Queue is a linear list of element in which deletion of an element can take place at one end, called "front" and insertion can take place at other end called "Rear"

The first element in a queue will be first one to be removed from the list.

Queue are also called FIFO (First In First out).



Basic Operations:

* Enqueue
* Dequeue

Application of queue:

*. Serving Requests on a single shared resources.

Eg: printer, CPU task scheduling (FIFO)

* In real life, call Center Phone Systems (People should wait on hold service member is free

→ Handling of interrupts in real time systems

The queue as an ADT:

A queue of type T is a finite sequence of elements with the operations:

* MakeEmpty(Q): To make Q as an empty queue.
* IsEmpty(Q): check Q is empty, if Empty return True or false.
* ISFULL(Q): check Q is full. if full return True or false.
* EnQueue(Q,X): insert X at rear end of the queue. if and only if Q is not full.
* DeQueue(Q): delete item from front of the queue if Q is not empty.
* Traverse(Q): To read entire queue that is display the content of the queue.

Sample operations

→ Queue q;

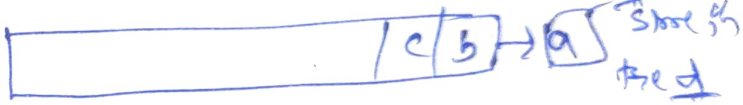
→ enqueue (q, 'a');

→ enqueue (q, 'b');

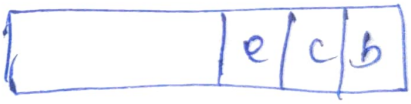
→ enqueue (q, 'c');

→ d = getFront(q);

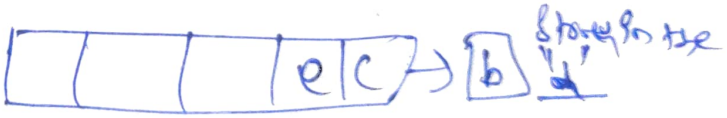
→ dequeue(q);



→ enqueue(q, 'e');



→ dequeue(q);



The Implementation of Enqueue:

```
Int enqueue (int data)
{
    if(isfull())
        return 0;
    rear = rear+1; [Inserting.
    queue[rear] = data;
    return 1;
}
```

Implementation of dequeue:

```
int dequeue()
{
  if (isempty())
    {
      return 0;
    }
  int data = queue[front];
  front = front+1;
  return data;
}.
```