

08/03/2024

|          |  |
|----------|--|
| Page No. |  |
| Date     |  |

## ## Day 11 of DSA

### ## Tasks:

Check  
Box

Queue



Queue Operations : Enqueue, dequeue



Queue Implementation : Using Arrays,  
Linked list.



Implement enqueue, dequeue operations  
using arrays

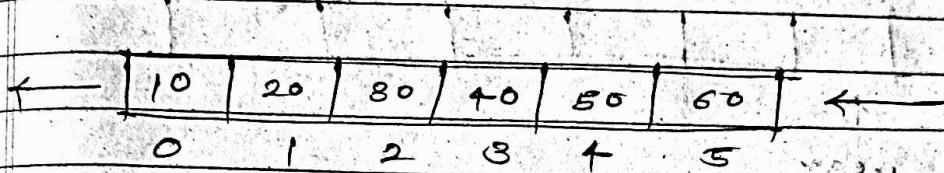


Implement enqueue, dequeue operations  
using linked list.

\*

Queues &amp; stacks

First In First Out.



\*

Operations:

(1) push

(2) pop

(3) size

(4) isempty

\*

STL :-

queue&lt;int&gt; q;

push → q.push(10);

pop → q.pop();

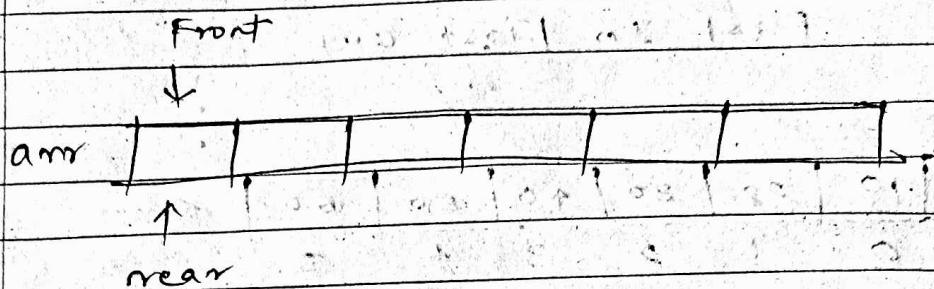
size → q.size();

empty → q.empty();

q.front() = For first element.

OK

## Implement Queue using Array:-



① Push :-

if (arr == full) {

    queue is full

} else

    arr [rear] = element;  
    rear++

}

② pop :-

    if (Front == rear) {

        Front = 0;

        rear = 0;

    } else {

        wastage of memory

        queue is empty.

} else

    arr [Front] = -1;

    Front++

}

(9) front :-

$\text{if } (\text{empty}) \rightarrow \text{return -1}$   
 else

arr[Front]

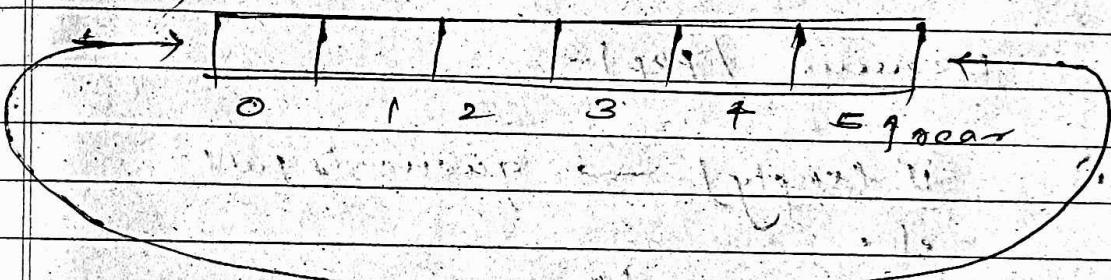
Time complexity :  $O(1)$ 

\*

## Circular Queue

Front

↓



(1)

push :-

$\text{if } (\text{Full}) \rightarrow \text{queue is full;}$

(Front == 0 && rear == size - 1)

rear = (Front + 1) % (size - 1)

ii) First element :-

$\text{if } (\text{Front} == -1)$

{

Front == rear == 0

arr[queue] &lt; data

}

II)

if ( $\text{rear} == \text{size} - 1$ ) { if ( $\text{front} == 0$ )

{  $\text{rear} = 0$

$\text{arr}[\text{rear}] = \text{data}$  } ;

}

else {

{

$\text{rear}++$  ;

$\text{arr}[\text{rear}] = \text{data}$  ;

}

## \* Dequeue / pop) -

(1) if (empty)  $\rightarrow$  queue is full.

else

$\text{Front} = -1$  ;

(2) single element in queue.

if ( $\text{front} == \text{rear}$ )

{

$\text{Front} = \text{rear} = -1$  ;

}

(3) if ( $\text{front} == \text{size} - 1$ ) { if ( $\text{front} = 0$ ) }

(4)

$\text{Front}++$  ;

\* I/P restricted queue

i/p on one side

o/p. on both sides

push back ()

pop front ;

pop back ()

\* output restricted queue

i/p on both sides

o/p. on one side

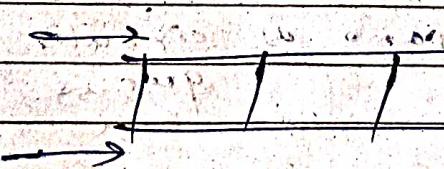
push-front

push-back

pop-front

\* Doubly ended queue

push-front



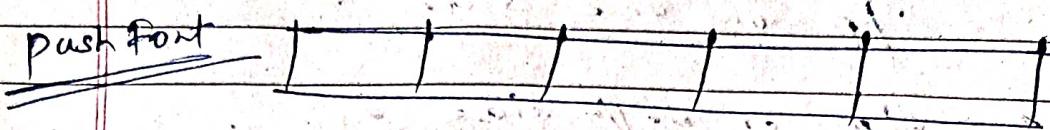
push-back

pop-front

pop-back

## Implementation of Queue

~~push front~~



$\text{front} = -1$

$\text{rear} = -1$

(I)

push front 1

① if Full  $\rightarrow$  queue is full

② single element.

$\text{Front} := -1$

$\text{front} = \text{rear} = 0$

③ if ( $\text{Front} = 0$ )  $\rightarrow \text{Front} = n-1$

④ normal flow  $\Rightarrow \text{Front} --;$

(II) push rear

(III) pop front

same as creator  
queue

(IV)

pop back

① empty queue is empty

② single element  $\rightarrow \text{front} = \text{rear} = -1$

③ cyclic nature if  $\text{rear} = 0$

$\text{rear} = n-1;$

④ normal flow

$\text{rear} --;$