DSA

INTERNSHALA

Stacks and Queues

Fundamentals of stack

A stack is a data structure which follows the principal of LIFO (Last in First Out).

The element inserted in the last is the one to be removed first.

There is only one end in a stack – Top

- If you want to insert a new element it will only on the top.
- If you want to remove an element it can only be from top.

Stacks can be represented using arrays and linked list.

- Inserting an element is called pushing
- Removing of the element is called popping
- If value of top is equal to the array size and user tries to push a new element, it is called overflow error.
- If value of top is 0, and user tries to remove an element, it is called underflow error.

[Algorithm for pushing an element into a stack]

```
1. Set top:= top + 1
2. Set stack[top]:=val
3. Return
```

[Algorithm for popping an element from a stack]

```
1. Set val:= stack[top]
2. Top:= top-1
3. Return
```

Fundamentals of Queues

A queue is a data structure that follows the principle of FIFO (first in first out).

There are two ends in a queue – front and rear

An element is always removed from the front

A new element is always inserted from rear.

[Algorithm for inserting an element in a Queue]

```
1. If (front = 1 and rear = N) or front= rear+1, then
   Write: Queue overflow error
   Return
   [end of if]
2. If front = NULL and rear = NULL, then
   Set front:= rear+1
   Else
   If rear=N, then
   Set rear:=1
   Else
   Set rear:=rear+1
   [end of if]
   [end of if]
3. Set queue [rear]:= val
4. Return
```

[Algorithm for deleting an element from a Queue]

```
1. If front = 0, then
   Write: Queue underflow error
   Retun
   [end of if]
2. Set val:= queue[front]
3. If front = rear, then
   set front:= rear:=0   [now queue is empty]
   else
   if front= N, then
   set front:=1
   else
   set front:=front+1
   [end of if]
   [end of if]
4. return
```