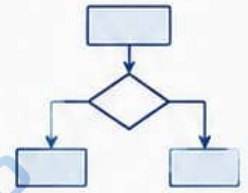
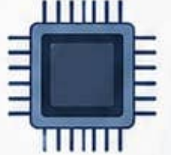


1010101010
0101010101
1010101010
0101010101



10110
01001
10101

Unit - II



Stack PUSH

POP

42

classmate

Date _____
Page _____

Stack -

A stack is a linear data structure in which the insertion and deletion of elements are performed from only one end, called TOP of stack.

LIFO Principle -

Stack follows the LIFO (Last In, First Out) principle.

This means that the element which is inserted last is the first to be removed.

Eg -

A stack of plates.

The plate placed on the top is removed first.

Memory Representation -

A stack can be represented by using two common methods -

1) Array

2) LinkedList

1) Array Representation

In this method stack is represented using linear array.

- TOP indicates the position of last inserted element
- Initially TOP = -1 (Stack is Empty)

2	30	← TOP
1	20	
0	10	

~~* PUSH & POP Operations~~

~~⇒ PUSH~~

* Operation on Stack-

Stack performs all operations at one end only called TOP, and follows the LIFO principle.

⇒ PUSH-

PUSH operation is used to insert an element into stack. Before inserting an element first you have to check stack overflow.

Algorithm:-

(Stack, TOP, MAX, ITEM)

Condition: Check for Overflow.

Step 1 - If $TOP == MAX - 1$,
print "Stack Overflow".
Exit.

Step 2 - Set $TOP = TOP + 1$

Step 3 - $Stack[TOP] = ITEM$

Step 4 - Exit

2) Eg-

If a Stack has element $[10, 20, 30]$
 $(TOP = 30)$ and you have to PUSH 40

i) Check if space is available

ii) Move TOP pointer to the next index

iii) Place 40 at the TOP

→ new stack → $[10, 20, 30, 40]$

$[TOP = \text{location of } 40]$

2) POP.

POP operation is used to remove the top element from the stack. For removing an element from stack you have to check the Underflow Condition first.

Algorithm - POP[STACK, TOP, ITEM].

Condition -

check for Underflow -

Step 1 - If $TOP == -1$,

→ print "Stack Underflow".
Exit.

Step 2 - Set $ITEM = STACK[TOP]$

Step 3 - $TOP = TOP - 1$

Step 4 - Exit.

Eg -

If stack has elements $[10, 20, 30, 40]$
($TOP = 40$) and you have to POP then

i) Check if stack is empty → No

ii) Access top element = 40

iii) Remove 40.

iv) Move TOP to the location of 30

v) Exit.

Stack Overflow -

Stack overflow condition occurs when a push operation is performed on a full stack.

Condition -

$$TOP = MAX - 1$$

Reason -

No free memory space available

Stack Underflow -

Stack Underflow occurs when a POP operation is performed on an empty stack.

Condition -

$$TOP = -1$$

Reason -

There is no element to remove from the stack.

Queue-

classmate

Date _____

Page _____

Queue-

A Queue is a linear data structure in which the insertion of an element is done at one end called REAR and deletion of element done from other end called as FRONT.

FIFO Principle- (First In First Out)

Queue follows FIFO Principle. This means the element which is inserted first is to be removed first.

Example-

Queue at a ticket counter.

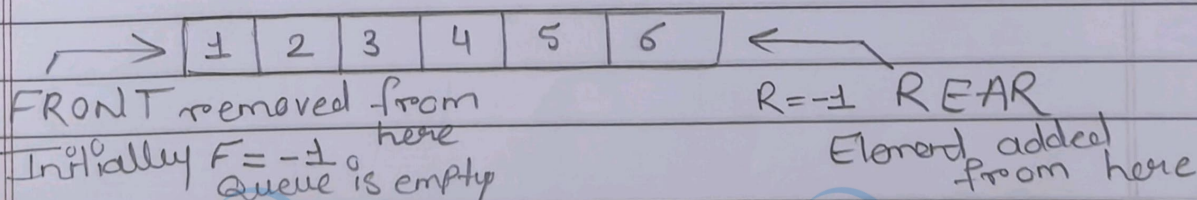
The person who come first is served first.

Queue Representation. by using Array.

In this method queue is represented by using Linear array.

Two variables are used :

- i) FRONT - points to the first element
- ii) REAR - points to the last element



Types of Queue-

Queue is a linear data structure that follows FIFO (First In First Out) principle.

Based on the way of elements are inserted and deleted, queue are classified into the following types.

1) Simple Queue / Linear Queue -

In simple queue insertion happens at the rear and deletion happens at the front.

Once the rear reaches the end of array, no more elements are added even if there is free space at front. (wasted space)

FRONT →

20	40	60	80	90
----	----	----	----	----

 — REAR

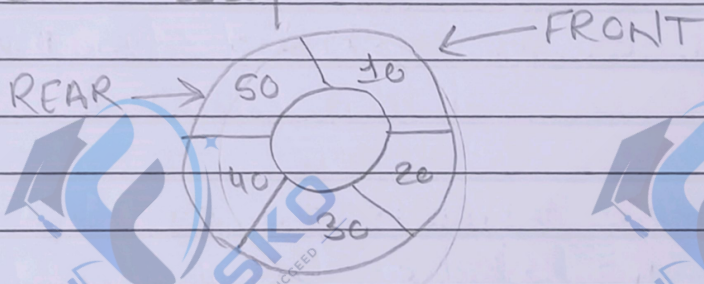
Dequeue removes first 20 then 40 and so on.

2) Circular Queue.

Circular Queue solves the problem of wasted space in a simple queue.

In Circular Queue the last position is connects back to the first position in a circular way.

Insertion and deletion happens in a circular way.



Eg- If the array of size 5.
 $\text{int } A[5] = \{1, 2, 3, 4, 5\};$

If we remove 1 and 2, front move forward

Now,

when we insert again the new element will be stored at beginning of the array.