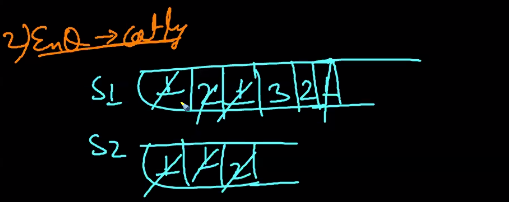
Implement Queue using stack:

Enqueue operation -> O(1)

Dequeue operation -> O(n)

Front operation -> O(n)

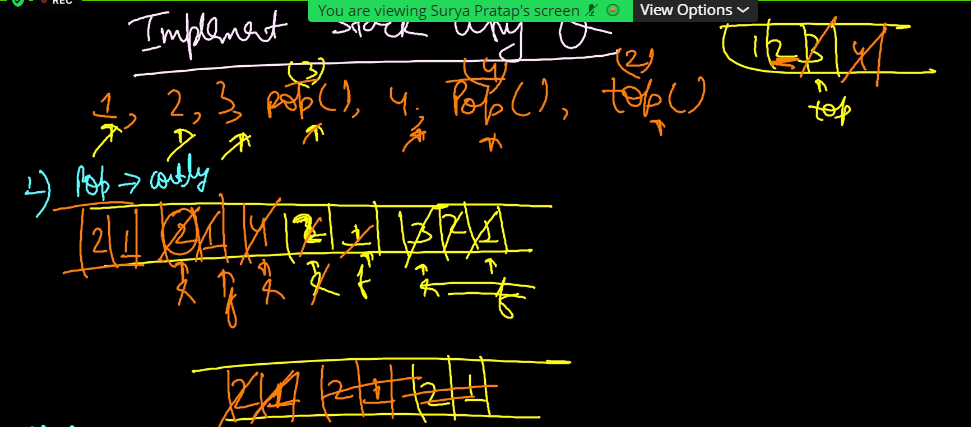
When we make Enqueue costly:



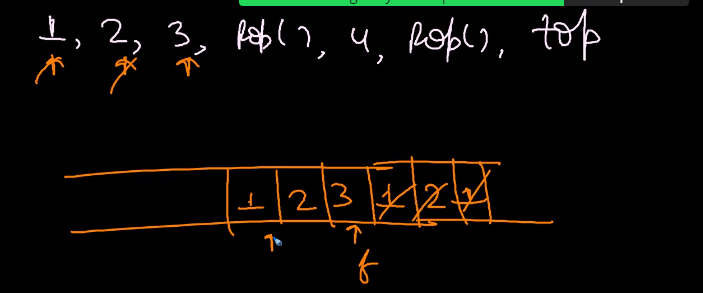
Here dequeue made O(1) and enqueue is o(n)

**Implement stack in queue;**

Pop Costly operation:



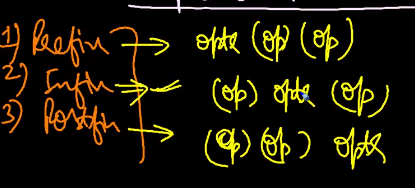
Push Costly operation:



Expression Conversions:

1. Prefix
2. Infix
3. Postfix

General expressions are infix expression.

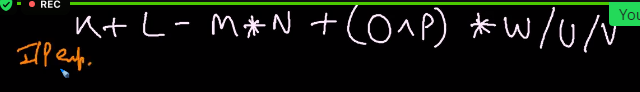


Expression Evaluation:

1. Presidency
2. Associativity only with infix expression.
3. Brackets only with infix expression

Only Presidency is there in Prefix and postfix

**Example;**

Input: 

First solve the highest presidence first.

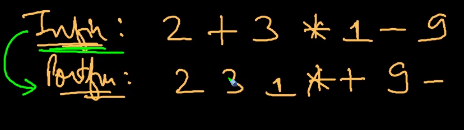
Let us take two stack for doing this.



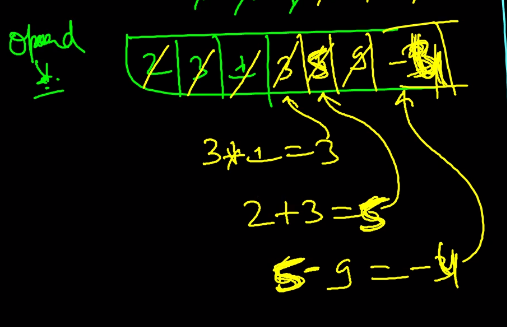
Another Example:



**Evaluation postfix expression:**

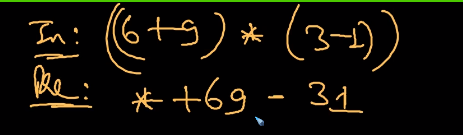
****

Push the operand till a operator is dectected and do the operation with 2 operands in the stack and repeat it till the end to get the calculated value from postfix expression.



It is just a postfix evaluation

**Prefix evaluation;**



Operand stack;

In here we need to push the operand from right to left.

