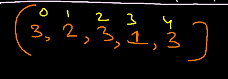
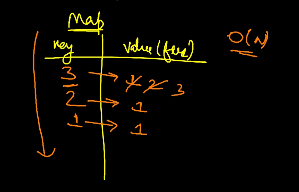
Moore’s Voting Algorithm: (majority element)



We can use map concept for find the maximum frequency of the element.



Step 1: find the candiate

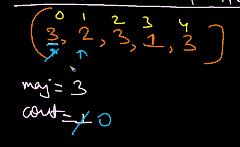
Step 2: verify the candidate as majority element.

The above the 2 steps gurantees the majority element.

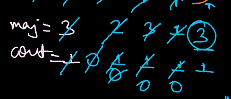
(Need the get the frequency of the element and check them wheather it is greater than n/2. If it is greater than n/2 then it is a majority element.)

Let us take first element as our majority element and start the process.

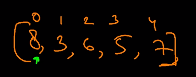
And count as 1.



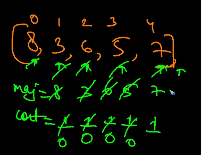
Restart the count to 0. After traversing with new element. At the end we get a element count as positive. Check the complete frequqncy in the array which is need to be more than n/2.



Example:

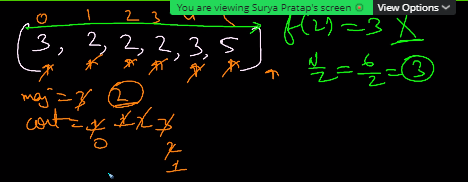


Assume the first element as majority element



Here n/2 > than the last total frequency of the element

**Another Example:**



Stack and Queue:

Implement Stack and Queue

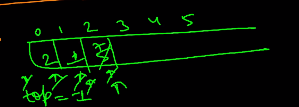
1. Array
2. Linked List

In CPP vector<int> is dynamical array

Array:

Top pointer is the last element position or top element of the stack.

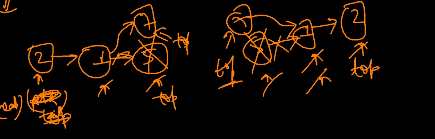
We need to increment the top value to insert the value and to remove the value we need to remove the top value, as in the stack



**Linked List:**

Same as array, it is connected as nodes.

As we traverse the head.. Likwise top pointer will be traversed through the nodes till the end.

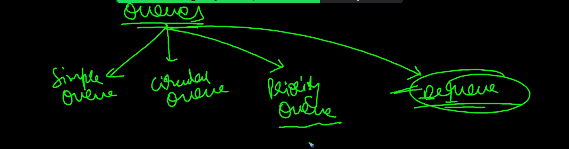


Recommended:

Array = stack

Queue -> linked list

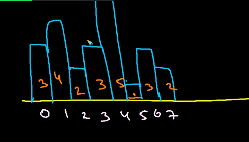
**Queue;**

****

**Use case of stacks:**

1. **Maintain the elements in order.**
   1. **ASC**
      1. **Maintain it based on priority**

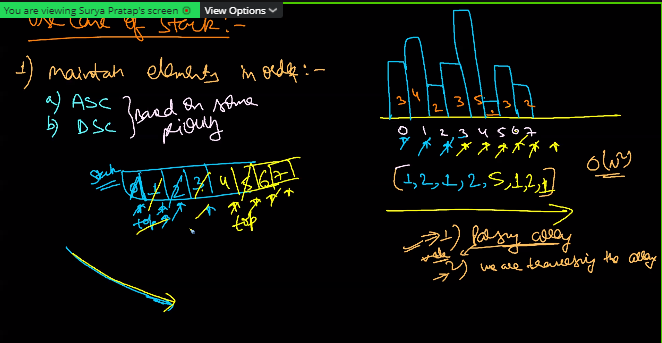
**Example;**

****

* 1. **DSC**
     1. **Maintain it based on priority**

Example:

Finding span or hiting graph from the previous.



Time Complexity: O(n)

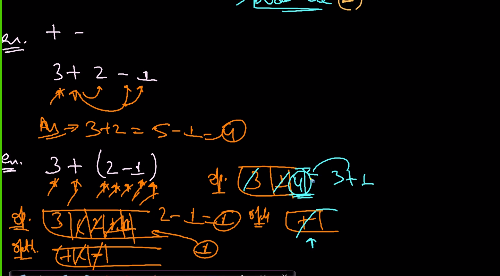
It is monotonically Decreasing stack

Stock span problem on Techdose.

1. Find previous greater element of all element of the array.
2. Find the next greater element of all element of the array.
3. **Maintain FIFO**
4. **Evalvuating Expression**
   1. **Associativity**
   2. **Presidence ( It is important)**

**For Example + and – has same presidemce and their associativity from left to right.**

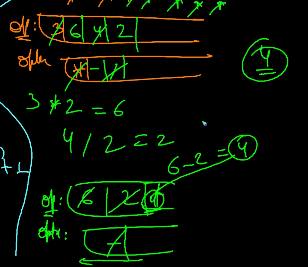
Brackets are used to change the presidence to our own way.



Doing operation of the expression using stack.

Example:





It is complex and there is a standard method.