**Sorting problem**

1. Check if array is sorted and rotated here

2.

**Array manipulation**

1. Move all zero to the end this is also a 2 pointer approach

Can take a look at the code here quite interesting here

2. and this is an interesting question here and then we keep moving forward here

Right!

**Heap problem**

1. Task scheduler

We pop element that has > 1 count, then add it back later on here

What happens if you use maxHeap?

This is O(log n) or O log( k)

**Stack problem here**

1. Next permutation problem

2. 132 pattern

**3. Make string great again**

Input: s = "leEeetcode"

Output: "leetcode"

**Memorable problems:**

**A special operation here**

**Contiguous array using map and others**

1. Keeping the index of the postion and subtracting the distance here, check out the problem contiguous array and then you will know
2. And then when you iteratre through more indexes, you do

maxLength = max(maxLength, i – mp[s])

**make sum divisible by k here**

1. Initialization of a hash map last with a key-value pair {0: -1} which tracks the modulus of the [prefix sum](https://algo.monster/problems/subarray_sum) and its index.
2. The question in place is make sum divisible by k here

Stack problem

1. Next greater element I, 2 and 3

First try the problem next largest element to the right here

1.

Binary search problem

1. Find local maxima of an array

**Sliding window problem**

1. Longest subarray of 1 after deleteing 1 elment

A sliding window problem with a bit of a manipulation here

**Minimum window substring problem here**

What did we learn from the minimum window substring problem here?

1. When using sliding window with map, you can use a hashmap to keep track of the character and the count

We can use the a count varaible to see if all the characters in the map has been found already here

String problem

1 Next permutation here

**Example 1:**

**Input:** nums = [1,2,3]

**Output:** [1,3,2]

This is very similar to next greater lexiographical sequence

A sub problem would be permutation in string here

String manipulation here

1.

Tree prbolem ‘

How does the tc usually go for binary tree problem?

Usually we can expect a O(n) if every tree node is visited here

Sum root to leaf numbers leetcode code

*The problem is if the root is root = [1, 2, 3]*

*The answer then becomes 12 + 13 => 25*

Tree problem here

1. Vertical order traversal leetcode

Return all the nodes in the same column basically here

-

And then the code here would become

**Type 2:  
using a stack to keep pointer**

Using stack to keep the running curruent Sum here

Stack = [ (root, sum - root.val),]

Binary tree longest consecutive sequence

Graph

1. Shortest binary with increasing matrix leetcode as well

Very simliar to rotten oragnes leetcode