Array

**Product of array except self**

1. There is the O(n) solution using the 2 passes here using 2 arrays here
2. There is also an O(1) appraoch here with the space instead of having 2 arrays here as said
3. This is quite interesting here and then we would have the solution here

**Question defuse the bomb here**

To decrypt the code, you must replace every number. All the numbers are replaced **simultaneously**.

* If k > 0, replace the ith number with the sum of the **next** k numbers.

**Input:** code = [5,7,1,4], k = 3

**Output:** [12,10,16,13]

**Explanation:** Each number is replaced by the sum of the next 3 numbers. The decrypted code is [7+1+4, 1+4+5, 4+5+7, 5+7+1]. Notice that the numbers wrap around.

Question

Boats to save people

Interesting problem

You are given an array people where people[i] is the weight of the ith person, and an **infinite number of boats** where each boat can carry a maximum weight of limit. Each boat carries at most two people at the same time, provided the sum of the weight of those people is at most limit.

**Palindrome 2 is when you**

Given a string s, return true *if the*s*can be palindrome after deleting at most one character from it*.

1. Take k of each character from left and right also a very good question here

**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

Stack problem

1. Next greater element I, 2 and 3

2. Remove duplicate letters

You must make sure your result is **the smallest in lexicographical order**

 among all possible results.

**Example:**

1. **Input:** s = "bcabc"

**Output:** "abc"

And then the above is it

3. 132 pattern:

Find a number

Given an array of n integers nums, a **132 pattern** is a subsequence of three integers nums[i], nums[j] and nums[k] such that i < j < k and nums[i] < nums[k] < nums[j].

**Using the 2 array appraoch problem**

1. Maximum swap

Traverse from the back once to the front and then get sth, traverse from the front to the back the 2nd time and then get sth else this would be it

2. Candy problem (hard)

- the first iteration:

**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"

**Sorting problem**

**1. Check if array is sorted and rotated here**

**2.**

**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

String problem

Make string great again

1. **Input:** s = "leEeetcode"

**Output:** "leetcode"

**Explanation:** In the first step, either you choose i = 1 or i = 2, both will result "leEeetcode" to be reduced to "leetcode".

2. Next permutation here

**Example 1:**

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

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

This is very similar to next greater lexiographical sequence

Linkedlist problem

1. Qustion : insert a value into circular sorted linkedlist

* This teaches us how to use a current and a previous pointer to insert while traversing a list

Bit manipulation problem for the afternoon

1. Square root (x) here using binary search appraoch ehre

**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

**Tree problem here**

**Preorder**

**1.** The parent come first and then the left and then right

***1  
 / \  
 2 3  
 / /   
4 5 6  
 / \  
 7 8***

This will then print us

1, 2, 4,

3, 5, 7

8, 6

**How to** find the maximum on each row here

1. take a look at this this is also quite imporatnt

2. You can try writing this out and seeing how this gets affected.

3. There is a template for this

1. Vertical order traversal leetcode

Return all the nodes in the same column basically here

Question 3:

**Binary tree longest consecutive sequence   
using a stack to keep pointer**

Using stack to keep the running curruent Sum here

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

**Bit manipulation**

1. What happens when you right shift sth by 1 n>> =1

This is equivalent to integer division by 2

1. n && 1 = 1

if it’s 1 it means n is an odd number here end of discussion

**Graph**

What are the 2 ways to find the shortest path in a graph?

1. There is 1 way where we add the distance to the queue as you traverse

Bfs\_matrix

2. There is another way where you iterate each level of the queue, and then keep trying

Bfs\_matrix\_2

1. Shortest binary with increasing matrix leetcode as well

Very simliar to rotten oragnes leetcode