**Leet Code Problem Set 2**

### ****1. 3Sum – Problem #15****

Find all unique triplets in the array which gives the sum of zero.

### ****2. Longest Substring Without Repeating Characters – Problem #3****

Given a string, find the length of the longest substring without repeating characters.

### ****3. Group Anagrams – Problem #49****

Group anagrams together from a list of strings.

### ****4. Top K Frequent Elements – Problem #347****

Given a non-empty array, return the k most frequent elements.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Here’s a few hopefully helpful tips on solving the above:**

1. **3Sum – Problem #15**

Input: [-1, 0, 1, 2, -1, -4]

Step 1: Sort the array

=> [-4, -1, -1, 0, 1, 2]

Step 2: Fix one number at a time (say, i)

Then use two pointers: `left` (i+1), `right` (end)

Example:

i=-1 (index 1), left=0, right=2 → sum = -1+0+1 = 0 ✅

Move pointers to find other triplets

1. **Longest Substring Without Repeating Characters – Problem #3**

Input: "abcabcbb"

Sliding window approach:

Start with two pointers: left and right

Right expands → [a], [ab], [abc] → length = 3

When duplicate found (second 'a'), move left to skip it → [bca]

Update max length every step

Final max = 3

1. **Group Anagrams – Problem #49**

Input: ["eat", "tea", "tan", "ate", "nat", "bat"]

Sort each word:

- eat → aet

- tea → aet

- tan → ant

...

Group by sorted form:

{

"aet": ["eat", "tea", "ate"],

"ant": ["tan", "nat"],

"abt": ["bat"]

}

Output is the grouped values

1. **Top K Frequent Elements – Problem #347**

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

Step 1: Count frequency → {1:3, 2:2, 3:1}

Step 2: Use a max heap or bucket sort to pick top k

Result: [1,2] (most frequent)

===========================================================