001-099

001. Valid Anagram (LC242)

Given two strings **s** and **t**, return true if **t** is an anagram of **s**, and false otherwise.

Example 1:
Input: s = "anagram", t = "nagaram"
Output: true

Example 2:
Input: s = "rat", t = "car"
Output: false

Constraints:

- 1 <= s.length, t.length <= 5 * 104
- s and t consist of lowercase English letters.

002. Group Anagrams (LC49)

Given an array of strings **strs**, group the anagrams together. You can return the answer in **any order**.

Example 1:
Input: strs = ["eat","tea","tan","ate","nat","bat"]
Output: [["bat"],["nat","tan"],["ate","eat","tea"]]

Explanation:

- There is no string in strs that can be rearranged to form "bat".
- The strings "nat" and "tan" are anagrams as they can be rearranged to form each other.
- The strings "ate", "eat", and "tea" are anagrams as they can be rearranged to form each other.

Example 2:
Input: strs = [""]
Output: [[""]]

Example 3:
Input: strs = ["a"]
Output: [["a"]]

Constraints:

- 1 <= strs.length <= 104
- 0 <= strs[i].length <= 100
- strs[i] consists of lowercase English letters.

003. Contains Duplicate (LC217)

Given an integer array **nums**, return true if any value appears **at least twice** in the array, and return false if every element is distinct.

Example 1:
Input: nums = [1,2,3,1]
Output: true

Explanation:
The element 1 occurs at the indices 0 and 3.

Example 2:
Input: nums = [1,2,3,4]
Output: false

Explanation:
All elements are distinct.

Example 3:
Input: nums = [1,1,1,3,3,4,3,2,4,2]
Output: true

Constraints:

- 1 <= nums.length <= 105
- -109 <= nums[i] <= 109

004. Longest Substring Without Repeating Characters (LC3)

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

Example 1:
Input: s = "abcabcbb"
Output: 3

Explanation: The answer is "abc", with the length of 3.

Example 2:
Input: s = "bbbbb"
Output: 1

Explanation: The answer is "b", with the length of 1.

Example 3:
Input: s = "pwwkew"
Output: 3

Explanation: The answer is "wke", with the length of 3.
Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.

Constraints:

- 0 <= s.length <= 5 * 104
- s consists of English letters, digits, symbols and spaces.

005. Find the Index of the First Occurrence in a String (LC28)

Given two strings **needle** and **haystack**, return the index of the first occurrence of **needle** in **haystack**, or -1 if **needle** is not part of **haystack**.

Example 1:
Input: haystack = "sadbutsad", needle = "sad"
Output: 0

Explanation: "sad" occurs at index 0 and 6.
The first occurrence is at index 0, so we return 0.

Example 2:
Input: haystack = "leetcode", needle = "leeto"
Output: -1

Explanation: "leeto" did not occur in "leetcode", so we return -1.

Constraints:

- 1 <= haystack.length, needle.length <= 104
- haystack and needle consist of only lowercase English characters.