Smart Coding & Interview Series
Top-20 Basic Program
(Map & MultiMap Applications)

First, understand the solution building strategies and coding for the problems in LIVE/VIDEO session and then you apply those strategies discussed in LIVE/VIDEO session to solve the following problems. Use your favourite language(C/C++/Java/C#/Python/Scala) for coding.

1) Most Common Word: Given a paragraph and a list of banned words, return the most frequent word that is not in the list of banned words. It is guaranteed there is at least one word that isn't banned, and that the answer is unique. Words in the list of banned words are given in lowercase, and free of punctuation. Words in the paragraph are not case sensitive. The answer is in lowercase.

Example:

Input:

paragraph = "Bob hit a ball, the hit BALL flew far after it was hit."

banned = ["hit"]

Output: "ball"

***Source**: https://leetcode.com/problems/most-common-word/*

2) Isomorphic Strings: Given two strings *s* and *t*, determine if they are isomorphic. Two strings are isomorphic if the characters in *s* can be replaced to get *t*. All occurrences of a character must be replaced with another character while preserving the order of characters. No two characters may map to the same character but a character may map to itself. You may assume both s and t have the same length.

Example:

Input: s = "egg", t = "add"

Output: true

***Source:** https://leetcode.com/problems/isomorphic-strings/description/*

3) Contains Duplicate-II: Given an array of integers and an integer *k*, find out whether there are two distinct indices *i* and *j* in the array such that nums[i] = nums[j] and the absolute difference between *i* and *j* is at most *k*.

Example:

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

Output: true

***Source**: https://leetcode.com/problems/contains-duplicate-ii/description/*

Smart Coding & Interview Series
Top-20 Basic Program
(Map & MultiMap Applications)

4) Anagram Mappings: Given two lists A and B, and B is an anagram of A. B is an anagram of A means B is made by randomizing the order of the elements in A. We want to find an index mapping P, from A to B. A mapping P[i] = j means the i^{th} element in A appears in B at index j. These lists A and B may contain duplicates. If there are multiple answers, output any of them.

Example

Input: A = [12, 28, 46, 32, 50] and B = [50, 12, 32, 46, 28]

Output: [1, 4, 3, 2, 0]

Explanation:

As P[0] = 1 because the 0th element of A appears at B[1], and P[1] = 4 because the 1st element of A appears at B[4], and so on.

Source: *https://leetcode.com/problems/find-anagram-mappings/description/*

5) Shortest Word Distance-II: Design a class which receives a list of words in the constructor, and implements a method that takes two words *word1* and *word2* and return the shortest distance between these two words in the list. Your method will be called *repeatedly* many times with different parameters. *You may assume that word1 does not equal to word2, and word1 and word2 are both in the list.*

Example:

Assume that words = ["practice", "makes", "perfect", "coding", "makes"].

Input: word1 = "coding", word2 = "practice"

Output: 3

Source: *https://leetcode.com/problems/ shortest-word-distance-iii/description/*