

## Smart Coding & Interview Series

### Top-20 Basic Program (Adhoc Thinking)

---

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) Reverse Words in a String:** Given a string, you need to reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order.

**Example:**

Input: "algorithmica is your best guru"

Output: "urug tseb ruoy si acimhtirogla"

**Note:** In the string, each word is separated by single space and there will not be any extra space in the string.

**Source:** <https://leetcode.com/problems/reverse-words-in-a-string-iii/>

**2) Find All Anagrams in a String:** Given a string **s** and a **non-empty** string **p**, find all the start indices of **p**'s anagrams in **s**. Strings consists of lowercase English letters only and the length of both strings **s** and **p** will not be larger than 20,100. The order of output does not matter.

**Example:**

Input: s: "cbaebabacd" p: "abc"

Output: [0, 6]

**Explanation:**

The substring with start index = 0 is "cba", which is an anagram of "abc".

The substring with start index = 6 is "bac", which is an anagram of "abc".

**Source:** <https://leetcode.com/problems/find-all-anagrams-in-a-string/description/>

**3) Merge Sorted Arrays:** Given two sorted integer arrays *nums1* and *nums2*, merge *nums2* into *nums1* as one sorted array.

**Note:**

- The number of elements initialized in *nums1* and *nums2* are *m* and *n* respectively.
- You may assume that *nums1* has enough space (size that is **equal** to *m* + *n*) to hold additional elements from *nums2*.

**Example:**

Input: *nums1* = [1,2,3,0,0,0], *m* = 3

*nums2* = [2,5,6], *n* = 3

Output: [1,2,2,3,5,6]

**Source:** <https://leetcode.com/problems/merge-sorted-array/>

## Smart Coding & Interview Series

### Top-20 Basic Program (Adhoc Thinking)

---

**4) Remove Duplicates from Sorted Array:** Given a sorted array `nums`, remove the duplicates in-place such that each element appear only once and return the new length. Do not allocate extra space for another array, you must do this by modifying the input array in-place with  $O(1)$  extra memory.

**Example:**

Given `nums = [1,1,2]`,

Your function should return `length = 2`, with the first two elements of `nums` being 1 and 2 respectively. It doesn't matter what you leave beyond the returned length.

**Source:** <https://leetcode.com/problems/remove-duplicates-from-sorted-array/description/>

**5) Shortest Distance between Words:** Given a sorted array `nums`, remove the duplicates in-place such that each element appear only once and return the new length. Do not allocate extra space for another array, you must do this by modifying the input array in-place with  $O(1)$  extra memory.

**Example:**

Input: `words = ["practice", "makes", "perfect", "coding", "makes"]`

`word1 = "coding"`, `word2 = "practice"`, return 3.

`word1 = "makes"`, `word2 = "coding"`, return 1.

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