数据结构：

数组、队列、栈、链表（单向和双向链表）、二叉树、哈希表、字符串、树、图、堆（最大堆和最小堆）

算法：

二分查找、动态规划、分治思想、贪心算法、递归、快慢指针、深度优先遍历、广度优先遍历、快速排序、归并排序、计数排序、回溯、滑动窗口、双指针

评判标准：

可读性、时间复杂度、空间复杂度

语言：

c++、Python

选题：

面试精典150题

做题思路：

模仿GitHub cheat sheet

记录：

题目序号、考察的知识点、解题思路

1. 合并两个有序数组

题目信息：两个非递减顺序排列的整数数组，两个整数分别对应其元素数目，合并2到1中，结果1同样按照非递减顺序排列，注意合并后存储在1中，故1的初始长度为2者长度之和，初始空闲位置的元素均为0。可以存在空数组，但不同时空，结果最长200，数值在正负10^9，题目进阶要求时间复杂度O(m+n)

考察知识点：数组、排序、双指针

解题思路：从条件来看数值在int32表示范围内，数组长度最长200，最短1。第一反应可想到的方法是申请一个新的空间，遍历两个数组，将元素依次存储到新空间中。但是题目要求原位存储在nums1中，这种方法不可行。双指针从两个数组的结尾开始遍历，存储的时候是从最后一个元素开始往前存储。考虑边界条件，如果其中有一方是空的判断，数组是否遍历完，存储元素索引是否到达0。

代码模块：边界判断是否存在空的数组，主while遍历，在遍历的模块中优先判断是否遍历完。