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。

1. 移除元素

题目信息：给定一个数组nums和一个值val，原地移除所有数值等于val的元素，返回nums中与val不同的元素的数量。

考察知识点：数组、双指针

解题思路：需要审题返回值只会考虑前面与val不同的元素的数量。因此最简单的思路是使用快慢指针，快指针遍历一次数组，慢指针指向需要返回的前面与val不同的元素部分，如果快指针指向的值不是val那么就复制到慢指针指向的位置处。这种方法时间复杂度是n。采用双指针，会更快，头指针遍历将所有val置换到尾指针，当头指针超过尾指针遍历结束。

1. 删除有序数组中的重复项

题目信息：已知非严格递增排列的数组nums，原地删除重复出现的元素，结果每个元素只出现一次，输出删除后数组的新长度。注意元素的相对顺序应该保持一致 ，nums被改变后前k个元素为不重复出现的，其余元素与其数组大小不重要。

考察知识点：数组、双指针

解题思路：双指针，快慢指针，慢指针指向需要返回的部分，快指针指向遍历的数组。边界条件是只有一个元素时不做处理。

**总结：考察双指针有一个特征是原位操作，不考虑数组中除返回部分的其他部分**

1. 多数元素

题目信息：已知nums数组大小为n，返回其中出现次数大于n/2的元素；进阶：尝试设计时间复杂度为O(n)、空间复杂度为O(1)的算法解决此问题。

考察知识点：数组、哈希表、分治、计数、排序

解题思路：使用哈希表，空间换时间，将数据存入哈希表中，key值为数字，value为出现的次数，查找到出现次数满足条件的就返回，但注意存在一个边界条件，就是只有一个元素或者所有元素都只出现一次的情况，返回首元素。还有一种思路是排序后返回n/2位置的元素，这两种方法空间复杂度不能达到O(1)，后者时间复杂度最快nlog2n。