结合 LeetCode 75 的特点和您当前的进度，以下是一些非常重要的注意事项，希望能帮助您在接下来的刷题过程中事半功倍：

### 一、 心态与策略层面

**目标不是“刷完”，而是“掌握”** LeetCode 75 是一个精心挑选的集合，旨在覆盖面试中最核心、最高频的知识点。您的目标不应该是简单地把这75道题做完打勾，而是要通过这75道题，**彻底掌握其背后所代表的数据结构和算法思想**。

**理解 LeetCode 75 的结构** 这个题单是按主题组织的（例如：数组、哈希表、双指针、滑动窗口、二叉树等）。当您在做一个主题下的题目时，要特别注意**这个主题的通用解法和思想**。例如，在做“双指针”分类的题目时，要思考双指针的两种主要形式（快慢指针、左右对撞指针）分别适用于什么场景。

**不要轻易看答案，但别钻牛角尖**

**给自己设定时间限制**：一道题，根据难度，可以给自己规定 30-45 分钟的独立思考时间。在这段时间里，尽力去分析、尝试。

**寻求“提示”而非“答案”**：如果卡住了，不要立刻去看完整的代码。可以先看看题解的标题，了解它用了什么算法（比如“啊，原来这题是动态规划”），然后关掉题解，再自己尝试去实现。或者，只看文字思路部分，理解了之后自己去写代码。

**最后的手段才是看代码**：实在没思路，再去看代码。但看完后，**一定要自己不看提示，独立地把代码重新敲一遍**，确保你真的理解了。

### 二、 技术与方法层面

**先思后写，谋定而后动** 在写下任何代码之前，请务必弄清楚以下几点：

**题意理解**：输入是什么？输出是什么？有什么限制条件（Constraints）？比如数组长度、数值范围等。

**边界条件 (Edge Cases)**：考虑一下特殊输入，例如：空数组 []、null、只有一个元素的数组、数组中所有元素都相同等等。在纸上或注释里把这些情况写下来。

**思路构建**：您打算用什么数据结构？用什么算法？为什么这个方法可行？在脑中或纸上模拟一遍您的算法，确保它能处理一般情况和边界情况。

**时空复杂度分析是刚需** 对于您写出的每一个解法，都要养成习惯去分析它的：

**时间复杂度 (Time Complexity)**：O(n), O(n log n), O(n^2)?

**空间复杂度 (Space Complexity)**：O(1), O(n)? 这不仅是面试的必考点，也是衡量您解法优劣的核心标准。在提交代码后，可以看看官方题解或者高票答案，对比一下他们的复杂度，思考自己的解法是否还有优化空间。

**举一反三，触类旁通** 这是刷 LeetCode 最重要的能力之一。

**识别模式 (Pattern Recognition)**：当您解决了一个问题后，问问自己：这道题的核心思想是什么？它属于哪种模式？（例如：滑动窗口、二分查找、回溯）。

**关联同类题**：LeetCode 网站右侧通常有“相似题目”列表。做完一道题后，花点时间去看看那些相似题目，思考它们和当前这道题的联系与区别。LeetCode 75 的妙处就在于，很多题目都是某一类问题的经典母题。

**温故知新，复盘总结至关重要** 人的记忆是会衰退的。刷题如果不复习，等于白刷。

**精读优质题解**：做完一道题后，无论您是否做出来，都**强烈建议**去阅读一下官方题解和讨论区高票的题解。看看别人是怎么想的，有没有更巧妙的思路或者更简洁的代码。

**建立自己的笔记**：可以用 Notion、GitHub Repo 或任何你喜欢的工具。记录下每道题的核心思想、解题模板、易错点、以及可以优化的方向。

**定期回顾**：可以采用“艾宾浩斯遗忘曲线”的模式，比如 1 天后、2 天后、4 天后、一周后...回过头来重新做（或者至少回顾一下思路）之前做过的题。目标是做到“**看到题目，5分钟内就能清晰地说出最优解法和时空复杂度**”。

### 总结

您现在正处在一个黄金时期。养成以上这些好习惯，您的刷题效率和效果将会大大提升。请记住：

**慢就是快**。

花 2 个小时深入理解一道经典题目，远比 2 小时内“打卡”式地刷掉 5 道题要收获得多。