

1 - 1 高效使用算法训练营

三分看视频理解、七分练习

- 视频：

最佳办法：1.5~2.0 倍数播放、难点（暂停+反复）

最差办法：类似看美剧一样，原速看完，仅一遍

- 练习：

Leetcode (力扣)

初期建议每天 3 - 5 题，以后每天 10题+（注：老题~90%）

摆正心态：承认其复杂性

科学做题：高频题

1. 数据结构和算法有其客观存在的复杂度
2. 一遍或两遍不理解，很正常（坚持五毒神掌）
3. 记忆（脑图）+反复（五毒神掌）是最为有效的方法

五毒神掌：即五遍刷题法，强调练习算法题要过数遍，算法题不仅是做一遍通过而已，而是通过刻意联系达到对算法熟练的效果。

误区

1. 什么都要追求“理解”；
2. 只看书不动手,忘记践行五毒神掌（有规律地过遍数）；
3. 谨记“通过”只是开始，关键要看高票代码和高质量题解（先BFS、再DFS）；
4. **最大误区：LeetCode 题目只做一遍！**

纠正：

1. 对于初学编程的人来说，无论年龄多大、智商多高，在编程语言相关的知识面前，其实依然是婴儿。放低身段，用鹦鹉学舌的姿势，不“理解”，别强求，只要能够依葫芦画瓢地去解题、考试，就够了；
2. 至于“理解”，交给时间。

摒弃“旧”习惯 - 最重要

- 不要死磕（传统方式）

五毒神掌（敢于放手、敢于死记硬背代码）

不懒于看高手代码（高赞题解、评论）

- 最佳方式：

五分钟想不出来，直接看题解或者高票代码，用五毒神掌变成自己的东西。这个过程，会觉得自己很菜甚至有点自卑，但是又借势而起的感觉。

- 最差方式：

看到题目想自己单挑一下，不结束外部帮助自己解决；以为15-30分之可以搞定，谁知道死磕了 2-3小时或者一晚上，终于“通过”。

筋疲力竭，没经历学习高票程序答案，就开始做下一题（或放弃）。

精通一个领域的步骤

- Chunk it up
- Deliberate Practicing
- Feedback

Chunk it up

It is important to view knowledge as sort of a semantic tree -- make sure you understand the fundamental principles.

- 切碎知识点
- 庖丁解牛
- 脉络连接

Deliberate Practicing

- 刻意练习 - 过遍数（五毒神掌）
- 练习缺陷、弱点地方：不舒服、不爽、枯燥
- 职业化运动
 - 基本功是区别业余和职业选手的根本
 - 基础动作的分解训练和反复练习

Feedback

- 即时反馈
- 主动反馈
 - 高手代码（GitHub，LeetCode，etc）
 - 第一视角直播
- 被动式反馈（高手给你指点）
 - Code Review
 - 教练看你打，给你反馈

切题四件套

- Clarification
- Possible Solutions :
 - 把所有可能的结果首先都过一遍，比较不同方法的时间复杂度和空间复杂度
 - Compare (Time/ Space)
 - Optimal (加强)

- Coding（多写）：只做一遍，是最大的误区；
- Test Cases

五遍刷题法（五毒神掌）

- 第一遍：
 - 5分钟，读题 + 思考；
 - 最多15分钟内，没思路的话就直接看解法；
 - 要看多解法，比较解法优劣；
 - 背诵、默写好的解法。
- 第二遍：
 - 马上自己写 → LeetCode 提交；
 - Debug 修改 → 提交；
 - 多种解法比较、体会 → 优化！
- 第三遍：
 - 过了一天后再重复做题；
 - 不同解法的熟练程度 → 专项练习；
- 第四遍：
 - 过了一周，反复回来练习相同题目；
- 第五遍：
 - 面试前一周恢复性训练。

建议：

- 语言基础加固：
 - `i+1, i++, str+"(", str+")"`
 - Java 为例：全局、局部、栈变量、堆变量、参数传递（传值、传ref）
- 工具翻新 (vscode/idea/google/iterm/...)
- 打字练习 [typing.io](https://www.typing.io/) 、 [SpeedCoder](https://www.speedcoder.com/)

笔记时间：2021-03-24 总结《算法训练营 25期》谭超专栏 --- Benjamin Song