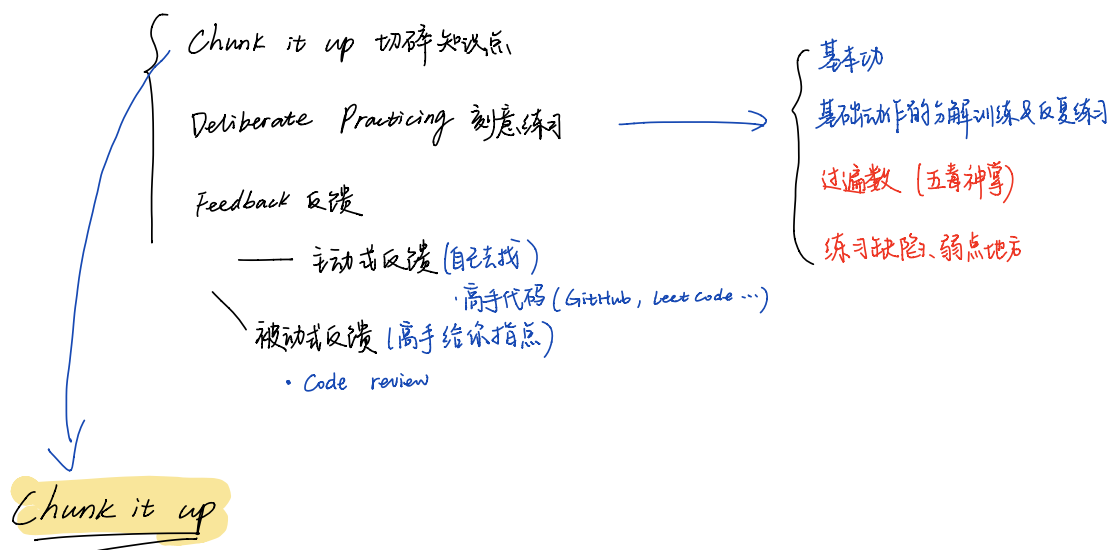


- ☆ {
 - 预习 - 基础课以自己预习和查看
 - 课堂互动 - 跟着老师一起思考、回答问题
 - 课后作业 - 按照切题办法

期待效果

- 对于算法数据结构的理解 · 达到顶尖级别
- good at - 一线互联网公司面试
- Leet Code 300+ 的积累 | 多多练习

精通一个领域



一维数据结构

基础: 数组 Array (string), 链表 Linked list

☆ 高级: 栈 stack, 队列 queue, 双端队列 deque, 集合 set

特点: 先进后出 先进先出

映射 map (hash or map) ...

补充特点

二维数据结构

基础: 树 Tree, 图 Graph

高级: (在基础的程度上加了很多特殊判断和约束条件)

- 二叉搜索树 binary search tree 根节点 $>$ 它的左子树且 $<$ 它的右子树的部节点
- 特殊结构: 红黑树 red-black tree,

AVL

- 堆 heap
- 并查集 disjoint set
- 字典树 Trie

特殊数据结构：用于工程中特定的情景

- 位运算 Bitwise
- 布隆过滤器 Bloom Filter

(常考) • LRU Cache 缓存

算法

基石

- If-Else, Switch → branch (跳转语句)
- For, while loop → Iteration (循环)
- 递归 Recursion (Divide & Conquer, Backtrace) (函数自己调用自己)

• 搜索 Search = 深度优先搜索 Depth First Search (DFS)

广度优先搜索 Breadth First Search (BFS)

启发式搜索 A*

.....

- 动态规划 Dynamic Programming
- 二分查找 Binary Search
- 贪心算法 Greedy
- 数学 Math, 几何 Geometry
- 排序算法

注意：
在头脑中回忆每种算法
的思想和代码模板

刷题技巧

四步系统化的思考题目的方式 刷题四件套

- 看到题目后
- Clarification: 确保自己对这个题目的理解是正确的
 - Possible Solutions: 想所有可能的解法来解这个题目
 - [比较不同的想法, 它们的 **时间** 和 **空间复杂度**]
 - compare (time / space)
 - optimal (加强)
 - Coding (多写)
 - Test cases: 测试样例 - 列举几个

五遍刷题法 (五毒神掌)

第一遍 ① 5-10分钟 = 读题 + 思考

② (没有任何思路的话, 以及不知道怎么做 → 正常)

直接看解法 = !!! 多解法, 比较解法优劣

☆ ③ 背诵 - 默写好的解法

第二遍 ① (背诵 - 默写好解法) 马上自己写

↓
Leet Code 提交 (Debug 直到通过)

② 多种解法比较 - 择优 → 优化!

Leet code 会显示
执行时间 & 内存消耗

第三遍

① 过了一天, 再重复做题

② 对自己不是特别熟练的那些题 → 专项练习

第四遍

① 过了一周后, 反复回来练习相同题目

② 不熟练的题目 → 专项练习

第五遍

① 面试前一个星期恢复训练