

极客大学算法训练营

期末复习

覃超

Sophon Tech 创始人，前 Facebook 工程师

数据结构

- 一维：
 - 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, 映射 map (hash or map), etc
- 二维：
 - 基础：树 tree, 图 graph
 - 高级：二叉搜索树 binary search tree (red-black tree, AVL), 堆 heap, 并查集 disjoint set, 字典树 Trie, etc
- 特殊：
 - 位运算 Bitwise, 布隆过滤器 BloomFilter
 - LRU Cache

时间复杂度

Data Structure	Time Complexity								Space Complexity
	Average				Worst				Worst
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
<u>Array</u>	$\theta(1)$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$
<u>Stack</u>	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$
<u>Queue</u>	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$
<u>Singly-Linked List</u>	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$
<u>Doubly-Linked List</u>	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$	$\theta(n)$	$\theta(1)$	$\theta(1)$	$\theta(n)$
<u>Skip List</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n \log(n))$
<u>Hash Table</u>	N/A	$\theta(1)$	$\theta(1)$	$\theta(1)$	N/A	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$
<u>Binary Search Tree</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$
<u>Cartesian Tree</u>	N/A	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	N/A	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$
<u>B-Tree</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$
<u>Red-Black Tree</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$
<u>Splay Tree</u>	N/A	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	N/A	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$
<u>AVL Tree</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$
<u>KD Tree</u>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$

• <https://www.bigocheatsheet.com/>

算法

- If-else, switch —> branch
- for, while loop —> Iteration
- 递归 Recursion (Divide & Conquer, Backtrace)
- 搜索 Search: 深度优先搜索 Depth first search, 广度优先搜索 Breadth first search, A*, etc
- 动态规划 Dynamic Programming
- 二分查找 Binary Search
- 贪心 Greedy
- 数学 Math , 几何 Geometry

注意：在头脑中回忆上面每种算法的思想和代码模板

脑图

- <https://naotu.baidu.com/file/b832f043e2ead159d584cca4efb19703?token=7a6a56eb2630548c>
- <https://naotu.baidu.com/file/0a53d3a5343bd86375f348b2831d3610?token=5ab1de1c90d5f3ec>

化繁为简的思想

1. 人肉递归低效、很累
2. 找到最近最简方法，将其拆解成可重复解决的问题
3. 数学归纳法思维

本质：寻找重复性 —> 计算机指令集

学习要点

- 基本功是区别业余和职业选手的根本。深厚功底来自于 — 过遍数
- 最大的误区：只做一遍
- 五毒神掌
- 刻意练习 - 练习缺陷弱点地方、不舒服、枯燥
- 反馈 - 看题解、看国际版的高票回答

经典习题

爬楼梯、硬币兑换

括号匹配、括号生成、直方图最大面积、滑动窗口

二叉树遍历、分层输出树、判断二叉排序树

股票买卖、偷房子、字符串编辑距离、最长上升子序列、最长公共子序列

异位词（判断和归类）、回文串（最大回文串）、regex和通配符匹配

高级数据结构（Trie、BloomFilter、LRU cache、etc）

五毒神掌

第一遍：不要死磕，要看代码学习（一定要看国际版的高票回答）

第二遍：自己写

第三遍：24小时后

第四遍：一周后

第五遍：面试前

面试技巧

1. Clarification: 明确题目意思、边界、数据规模
2. Possible solutions: 穷尽所有可能的解法
 - compare time/space
 - optimal solution
3. Coding: 代码简洁、高性能、美感
<https://shimo.im/docs/rHTyt8hcpT6D9Tj8>
4. Test cases

恭喜完成训练课程

师傅领进门，修行在个人

Commencement

Steve Jobs: https://www.youtube.com/watch?v=Hd_ptbiPoXM