

手把手带你搞定 10 道大厂算法面试题

覃超

互联网大厂面试的形态

- 头条面经解读

<https://www.bilibili.com/video/BV14z411b7sT>

- leetcode-cn.com 论坛

leetcode.com

<https://www.1point3acres.com/bbs/> （一亩三分地）

基础知识

- 语言 / 多并发 / 数据库 + 缓存 / 其他
Google → 好的源: github、掘金、知乎、程序园、csdn

- 以 java 为例

<https://github.com/Snailclimb/JavaGuide>

<https://github.com/CyC2018/CS-Notes>

<https://juejin.im/entry/58f1d35744d904006cf14b17>

算法“难题”的分析

解题思路 双指针移动，对比元素解题。当遇到重复元素时，左指针不动，右指针继续移动。关键在于理清遇到不相等元素时，左指针移动且其对应元素值的问题。（建议图解） 代码

每日1题
6月热题来袭!

每天0点题库首页置顶
坚持刷题，持续优秀，很可!

参与竞赛，便有机会赢取
奖品



在力扣商店使用你的力扣积分





70 天彻底攻克算法，斩获大厂 Offer

6 月 22 日将永久涨价至 ¥1999 起

题目一

- <https://leetcode-cn.com/problems/regular-expression-matching/>

特点

- 没思路
- 思维复杂度貌似非常高
- 一般是 分治 或者 动态规划 (DC ~ DP)

如何思考?

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

难题的分析

- <https://leetcode-cn.com/problems/climbing-stairs/>
- <https://leetcode-cn.com/problems/coin-change>
- <https://leetcode-cn.com/problems/decode-ways/>
- 寻找重复性 —> 分治以及DP的思路和想法

例题集合

- <https://leetcode-cn.com/problems/all-possible-full-binary-trees/>
- <https://leetcode-cn.com/problems/partition-array-for-maximum-sum/>
- <https://leetcode-cn.com/problems/burst-balloons/>
- 寻找重复性 —> 分治的思路和想法

例题集合

- <https://leetcode-cn.com/problems/generate-parentheses/>
- <https://leetcode-cn.com/problems/combinations/>
<https://leetcode-cn.com/problems/permutations/>
等

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



70 天彻底攻克算法，斩获大厂 Offer

6 月 22 日将永久涨价至 ¥1999 起

算法模板 + 脑图总结

- 图形化记忆和学习（脑图） + 费曼学习法
- 动手练习（五毒神掌）
- 班级、微信群、战友、人脉资源

过遍数 —> 脑图记忆

- 书先是越读越厚，然后越读越薄
- 看山是山，看山不是山，看山還是山
- 内功浑厚：重剑无锋、大巧不工



70 天彻底攻克算法，斩获大厂 Offer

6 月 22 日将永久涨价至 ¥1999 起

问题环节