

【第一章：C++后台开发学习路线】第3节：算法刷题提升路线

已购

来自【《收割BAT：C++校招学习路线总结》】 | 214 浏览 | 0 回复 | 2019-11-25



特立独行MVP



专栏作者

+关注

前言

相信很多人和我一样，都是非科班出身(计科/软院)，没有在最开始的时候接触最基础的算法课程和训练。我一开始也很苦恼，在硕士阶段才来弥补，一方面面对科研的压力平时时间很紧迫，另一方面不知道有哪些合适的资源可以去提升自己，往往浪费很多时间在找书找课上。网上的公开课很多很多，算法教程又是一大堆。同时网上刷题的OJ那么多，题目又是海量，到底该怎么入手？本文就将从我个人的痛点切入，介绍下我认为的比较好的算法学习策略，在本节最后我也对我秋招遇到的算法题做了一个总结。

痛点分析

个人总结下来，初期学习算法有以下几个痛点：

- 1.编程能力不足。初期由于编程能力不足会导致眼高手低，看着都会，写起来很费劲。
- 2.过于心急。很多同学在早期就想进入刷题阶段，算法基础还不扎实，题目类型都不熟悉，匆忙刷题只会打击自己的自信心。
- 3.效率低下。比如一开始就去看《算法导论》这样的大而全的书，会导致时间成本太高，效率很低。
- 4.选择困难。市面上太多资源，算法课，OJ，如何选择成为初期最大的困难。
- 5.定位错误。应对秋招的算法要求和算法竞赛的要求是不一样的，很多人缺乏明确的定位。
- 6.缺乏策略。很多同学因为没有策略，花了很多时间但是算法能力提升缓慢。不能用战术上的勤奋来掩盖战略上的懒惰。

算法能力指的是什么

算法能力不是指算法工程师的能力，算法能力是一项通用的基础能力，指的是对于基本算法和数据结构的了解程度和解决通用算法问题的编程能力。

所以：算法能力 = 数据结构 + 基础算法 + 编程能力 + 逻辑思维。

数据结构：对基本数据结构的设计思路，使用场景，性能特点的熟悉程度。

基础算法：对基础算法的设计思路，使用场景，复杂度的了解程度。

应对春招实习，秋招面试，应该具备什么算法水平

算法能力是比较容易量化的，一般来说对于大部分人，你能够解决LeetCode上全部easy题和大部分中等难度题，就可以应对大部分国内互联网公司的应试要求，再细化来说就是LeetCode 400-500题量水平。

如今国内互联网公司越来越喜欢考察应届生算法知识，体现在这几个方面：

- 1.增加手写算法题数目；
- 2.手写算法题的结果直接决定面试是否通过；
- 3.算法问题深度增加；

以我个人实际经历来说，大厂的面试尤其会在第2点体现更加突出。就阿里来说，阿里的技术面试会在正式开始前，面试官会通过邮件发一道算法题，只有应聘者在规定时间内完成手写算法题，才会开始后续面试内容。这意味着，算法题将成为第一道门槛。目前我面试过的公司当中算法比重较高的几家公司如下：字节跳动，依图，商汤，小马智行等等。

面试算法题一般不会很难，和算法竞赛的要求差别很大，如果简历上没有出现ACM等算法比赛获奖经历，面试官是不会特意为难。面试考察的是算法的基础知识和编程基本功，题目一般简短明确，容易理解，甚至是往年的原题，但是往往隐含边界分析，bug排查，所以需要谨慎小心。

常考算法知识点

- 排序：大部分要求能手写，并分析时间空间复杂度，以及稳定性分析
 - 快速排序，插入排序，冒泡排序，选择排序，堆排序，归并排序，基数排序
- 树：需要了解结构，遍历方式，前中后序遍历
 - 二叉树，红黑树，B树，B+树，Trie 树，胜者树，败者树
- 图：DFS，BFS，Dijkstra
 - 拓扑排序；并查集；最短路径；最小生成树
- 散列表：实现原理，扩容，效率
- 字符串：KMP，Manacher，
- 数组、链表、二叉树、队列、栈的各种操作（性能，场景）
- 其他算法：2分搜索，动态规划

算法学习路线

我把算法学习的过程分为2个阶段：

阶段1：课程学习阶段：这个阶段我建议先找一些比较靠谱的视频或者公开课去学会比较好。如果学校里的课还不错可以去上学校的专业课。如果不是计算机或者软院的学生，有条件的可以蹭课，

≡ 特立独行MVP

必修课	必修课	必修课
计算思维	计算机系统基础	计算机网络
程序设计基础	数据结构	软件质量保障
离散数学	算法设计与分析	数据库理论
数字电路	软件工程	高级程序设计
计算机图形学	概率论	编译原理
密码学	操作系统	网络安全

其中我认为和算法相关最重要的课程如下：

- 程序设计基础
- 数据结构
- 算法设计与分析

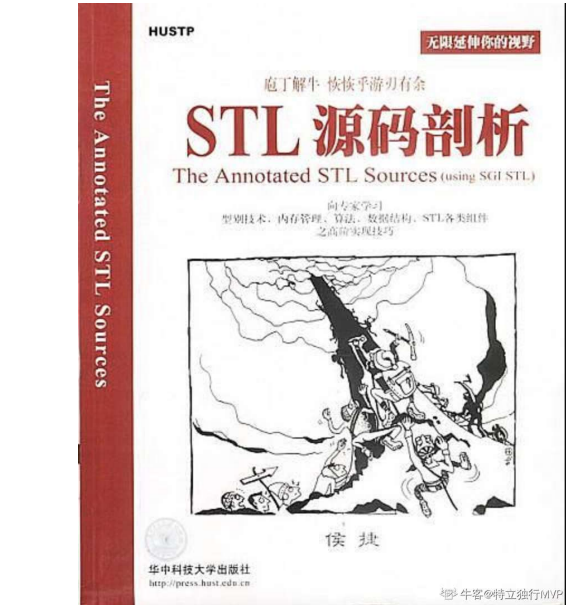
如果没有实际上课的条件我推荐网课的方式，下面是我学习的课程：

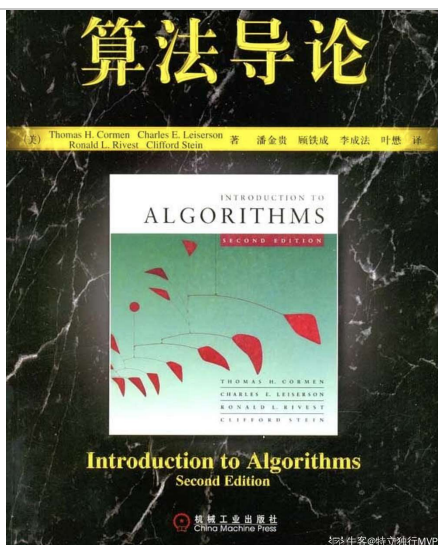
- 浙大的数据结构(入门用)
- MIT的算法导论课程
- 左神算法课

如果有条件可以去YouTube上看，如果没法也可以在B站上看，好处就是有字幕不用自己人肉翻译，同时多终端都可以看，比如地铁上可以拿出手机看，也是非常好的学习积累的方式~

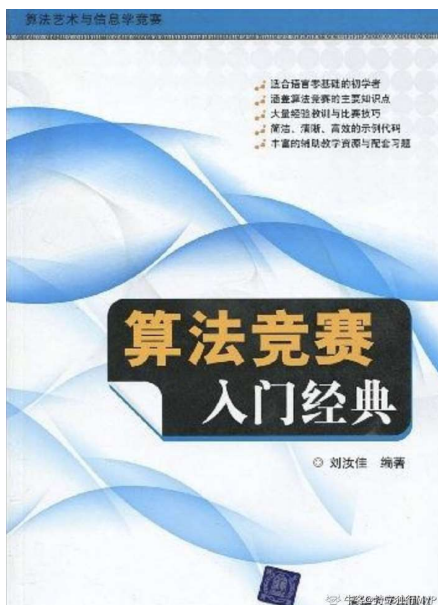
同时也建议大家在看完相应课程的时候看一看书去巩固知识和增强细节的记忆：

- 《STL源码剖析》：可以学习各种数据结构的C++实现





- 《算法竞赛入门经典》：动态规划讲的很好



如果你时间不够，那么牛客网左神的算法课你就不能错过了。非常系统和直接，都是干货。很多的面试题在左神的课上都有讲解。我个人建议有条件的同学可以刷1-2遍，尤其在春招实习之前。左神的课很系统，能够很好的把前面的知识点串联起来。

阶段2：刷题阶段：基础学完之后，就需要大量的刷题。我不推荐每天没日没夜的刷，我也曾经这样一天刷十几道，但是效率其实不高。我建议在有一定题量后每天写个1-2题就足够，难度leetcode中等。重要的是刷完题之后的积累！尤其是如何优化代码的思路和边界分析，不要去一味的追求题量。切记题目是永远也刷不完的。

刷题策略：

1.记录解法。我个人的做法是将每次写的题目和代码同步到Github上，通常1题会写1-2个解法，如果遇到特殊思路会将思路的关键词贴在commit上。一是方便自己查找，二是方便在日后遇到同样的题目时可以快速回忆解法。

2.限时做题。在刷题时需要给自己时间紧迫感，简单题10min，中等题25-30min。

刷哪些题

必刷

- 剑指offer： 建议刷2遍
- leetcode： 所有简单和中等题
- 左神算法课上的所有例题

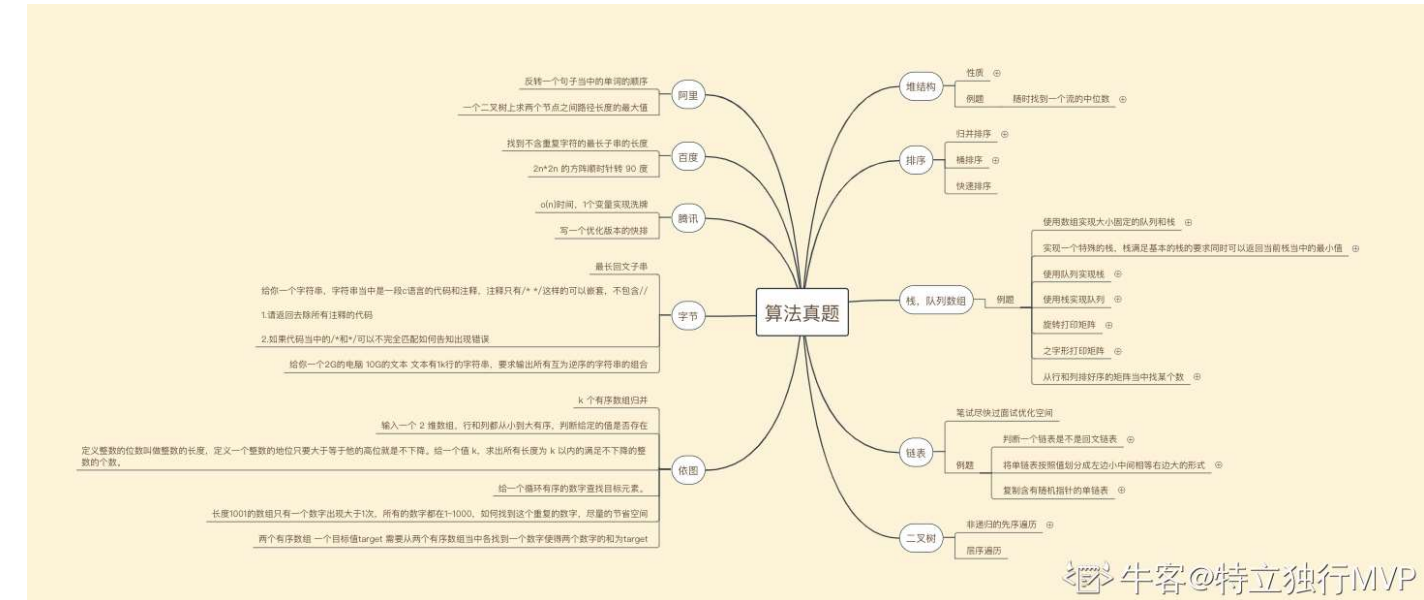
OJ个人推荐

- hihoCoder： 他会有一个每周周赛，都在周日，四道题。难度和代码量都还是挺大的，如果想去外企我建议多刷刷。
- lintcode： 为什么推荐呢，因为他有个CAT模式也就是天梯上分，写一题上一颗星每题有时间限制15分钟或者30分钟，提交次数限制3次，没过会惩罚加题。总之非常适合面试模式。我是刷的停不下来。
- codeforces： 这个就偏比赛一点，但是很多新题都是这上面改的所以值得做一做。

笔试算法题

笔试和面试是不同的，笔试追求的是尽快AC，在时间有限的情况下不需要在乎解法的优劣。笔试算法题安排的时间通常要半个小时到一个小时，所以难度会大一些。笔试算法题的内容比较灵活，很多会给很长一段场景描述，需要正确理解题意。不过有内推的情况下，笔试能避免就避免。

面试真题分享



特立独行MVP

- 2.阿里：CanTransform：如hit->hat, hat->cat, cat->fat。但是不能单步从hit->hog。设计算法判断给定输入能否在字典范围内经过任意次步骤完整转换；
- 3.阿里：一个二叉树上求两个节点之间路径长度的最大值；
- 4.百度：找到不含重复字符的最长子串的长度；
- 5.百度： $2n \times 2n$ 的方阵顺时针转 90 度；
- 6.商汤：按顺序打印二叉树的每一层的最后一个节点的值；
- 7.小马：将 1-n 这 n 个数按照 string 大小排序；
- 8.依图：给一个循环有序的数字查找目标元素；
- 9.依图：两个有序数组合并；
- 10.依图：k 个有序数组归并；
- 11.依图：输入一个 2 维数组，行和列都从小到大有序，判断给定的值是否存在；
- 12.猿辅导：求一棵树当中任意两个节点最长路径长度；
- 13.猿辅导：给你一个 list 都是父子节点的对，写个程序将其转换成森林，如果有环怎么处理？
- 14.腾讯： $O(n)$ 时间，1个变量实现洗牌；
- 15.字节：最长回文子串；
- 16.字节：给你一个字符串，字符串当中是一段c语言的代码和注释，注释只有/* */这样的可以嵌套，不包含//，请返回去除所有注释的代码，如果代码当中的/和/可以不完全匹配如何告知出现错误；

结语

在本节当中我主要介绍了算法提升的策略，在下一节当中将主要介绍后台开发方向的技能树。

感谢阅读，如果文中有任何错误欢迎你给我留言指出，也欢迎分享给更多的朋友一起阅读。

[举报](#)

收藏



赞 3

相关专栏



《收割BAT：C++校招学习路线总结》

19篇文章 | 95订阅

已订阅



没有回复

请留下你的观点吧~

发布


专栏推荐



《收割BAT：C++校招学习路线总结》

《收割BAT：C++校招学习路线总结》，专栏共计17节。专栏分为五大主要内容，包括后台开发学习...

19篇文章 | 95阅读

 牛客博客，记录你的成长

[关于博客](#) | [意见反馈](#) | [免责声明](#) | [牛客网首页](#)