

程序员练级攻略（2018）：理论学科

2018-06-14 陈皓



程序员练级攻略（2018）：理论学科

朗读人：柴巍 10'41" | 4.90M

进入专业的编程领域，算法、数据结构、网络模型、计算机原理等这样的计算机专业需要学习的理论知识是必须要学习的。下面我们先来讲述数据结构和算法。

数据结构和算法

算法是比较难学习的，而且学习“算法”是需要智商的。数组、链表、哈希表、二叉树、排序算法等一些基础知识，对大多数人来说是没什么问题的。但是一旦进入到路径规划、背包问题、字符串匹配、动态规划、递归遍历等一些比较复杂的问题上，就会让很多人跟不上了，不但跟不上，而且还会非常痛苦。是的，解决算法问题的确是可以区分人类智商的一个比较好的方式，这也是为什么好些公司用算法题当面试题来找到智商比较高的程序员。

然而，在很多时候，我们在工作中却发现根本用不到算法，或是一些基本的算法也没有必要实现，只需要使用一下第三方的库就好了。于是，导致社会上出现很多“算法无用论”的声音。

对此，我想说，算法真的很重要。我这 20 年的经历告诉我，无论是做业务还是做底层系统，经常需要使用算法处理各种各样的问题。比如，业务上我需要用算法比较两个数组中差异的布隆过

过滤器，或是在做监控系统时实时计算过去一分钟的 P99 统计时的蓄水池算法，或是数据库的 B+ 树索引，还有 Linux 内核中的 epoll 的红黑树，还有在做服务调度里的 " 背包问题 " 等都会用算法，真的是会本质上帮助到你，也是会让你非常有成就感的一件事。

虽然算法很难，需要智商，但我还是想鼓励你，这其中是有很多的套路是可以学习的，一旦学会这些套路，你会受益无穷的。

这里有几本书着重推荐一下。

- 基础知识。《[算法](#)》，是算法领域经典的参考书，不但全面介绍了关于算法和数据结构的必备知识，还给出了每位程序员应知应会的 50 个算法，并提供了实际代码。最不错的是，其深入浅出的算法介绍，让一些比较难的算法也变得容易理解，尤其是书中对红黑树的讲解非常精彩。其中，还有大量的图解，详尽的代码和讲解，也许是最好的数据结构入门图书。不好的是不深，缺乏进一步的算法设计内容，甚至连动态规划都未提及。另外，如果你觉得算法书比较枯燥的话，你可以看看这本有趣的《[算法图解](#)》。
- 理论加持。如果说上面这本书偏于实践和工程，而你看完后，对算法和数据结构的兴趣更浓了，那么你可以再看看另一本也是很经典的偏于理论方面的书——《[算法导论](#)》。虽然其中的一些理论知识在《算法》那本书中也有提过，但《算法导论》这本书更为专业一些，是美国计算机科学本科生的教科书。
- 思维改善。还有一本叫《[编程珠玑](#)》的书，写这本书的人是世界著名计算机科学家乔恩·本特利（Jon Bentley），被誉为影响算法发展的十位大师之一。你可能不认识这个人，但是你知道他的学生有多厉害吗？我例举几个，一个是 Tcl 语言设计者约翰·奥斯德奥特（John Ousterhout），另一个是 Java 语言设计者詹姆斯·高斯林（James Gosling），还有一个是《算法导论》作者之一查尔斯·雷斯尔森（Charles Leiserson），还有好多好多。这本书也是很经典的算法书，其中都是一些非常实际的问题，并以其独有的洞察力和创造力，来引导读者理解并学会解决这些问题的方法，也是一本可以改善你思维方式的书。

然后，你需要去做一些题来训练一下自己的算法能力，这里就要推荐 [LeetCode](#) 这个网站了。它是一个很不错的做算法训练的地方。现在也越做越好了。基本上来说，这里会有两类题。

- 基础算法题。其中有大量的算法题，解这些题都是有套路的，不是用递归（深度优先 DFS，广度优先 BFS），就是要用动态规划（Dynamic Programming），或是折半查找（Binary Search），或是回溯（Back tracing），或是分治法（Divide and Conquer），还有大量的对树、数组、链表、字符串和 hash 表的操作。通过做这些题能让你对这些最基础的算法的思路有非常扎实的了解和训练。对我而言，Dynamic Programming 是我的短板，尤其是一些比较复杂的问题，在推导递推公式上总是有思维的缺陷（数学是我的硬伤）。做了这些题后，我能感到我在动态编程的思路受到了很大的启发。

- 编程题。比如：atoi，strstr，add two nums，括号匹配，字符串乘法，通配符匹配，文件路径简化，Text Justification，反转单词等，这些题的 Edge Case 和 Corner Case 有很多。这些题需要你想清楚了再干，只要你稍有疏忽，就会有几个 case 让你痛不欲生，而且一不小心就会让你的代码写得又臭又长，无法阅读。通过做这些题，可以非常好地训练你对各种情况的考虑，以及你对程序代码组织的掌控（其实就是其中的状态变量）。

我觉得每个程序员都应该花时间和精力做这些题，因为你会从这些题中得到很大的收益。我在 Leetcode 上做一些题的代码在这——我的 [Github](#) 上，可以给你一些参考。

如果能够把这些算法能力都掌握了，那么你就有很大的概率可以很容易地通过这世界上最优的公司的面试，比如：Google、Amazon、Facebook 之类的公司。对你来说，如果能够进入到这些公司里工作，那么你未来的想像空间也会大得多得多。

最后，我们要知道这个世界上的数据结构和算法很多很多，下面给出了两个网站。

- [List of Algorithms](#)，这个网站罗列了非常多的算法，完全可以当成一个算法字典，或是用来开阔眼界。
- 还有一个数据结构动画图的网站 [Data Structure Visualizations](#)。

其它理论基础知识

下面这些书，基本上是计算机科学系的大学教材。如果你想有科班出身的理论基础，那么这些书是必读的。当然，这些理论基础知识比较枯燥，但我觉得如果你想成为专业的程序员，那么应该要找时间读一下。

- 《[数据结构与算法分析](#)》，这本书曾被评为 20 世纪顶尖的 30 部计算机著作之一，作者 Mark Allen Weiss 在数据结构和算法分析方面卓有建树，他在数据结构和算法分析等方面的著作尤其畅销，并广受好评，已被世界 500 余所大学用作教材。
- 《[数据库系统概念](#)》，它是数据库系统方面的经典教材之一。国际上许多著名大学包括斯坦福大学、耶鲁大学、德克萨斯大学、康奈尔大学、伊利诺伊大学、印度理工学院等都采用本书作为教科书。这本书全面介绍了数据库系统的各种知识，透彻阐释数据库管理的基本概念。不仅讨论了数据库查询语言、模式设计、数据仓库、数据库应用开发、基于对象的数据库和 XML、数据存储和查询、事务管理、数据挖掘与信息检索以及数据库系统体系结构等方面的内容，而且对性能评测标准、性能调整、标准化以及空间与地理数据、事务处理监控等高级应用主题进行了广泛讨论。
- 《[现代操作系统](#)》，这本书是操作系统领域的经典之作，书中集中讨论了操作系统的基本原理，包括进程、线程、存储管理、文件系统、输入 / 输出、死锁等，同时还包含了有关计算机安全、多媒体操作系统、掌上计算机操作系统、微内核、多核处理机上的虚拟机以及操作系统设计等方面的内容。

- 《[计算机网络](#)》，这本书采用了独创的自顶向下方法，即从应用层开始沿协议栈向下讲解计算机网络的基本原理，强调应用层范例和应用编程接口，内容深入浅出，注重教学方法，理论与实践相结合。新版中还增加了无线和移动网络一章，并扩充了对等网络、BGP、MPLS、网络安全、广播选路和因特网编址及转发方面的材料。是一本不可多得的教科书。
- 《[计算机程序的构造和解释](#)》，这本书也很经典，是 MIT 的计算机科学系的教材。这本书中主要证实了很多程序是怎么构造出来的，以及程序的本质是什么。整本书主要是使用 Scheme/Lisp 语言，从数据抽象、过程抽象、迭代、高阶函数等编程和控制系统复杂性的思想，到数据结构和算法，到编译器 / 解释器、编程语言设计。
- 《[编译原理](#)》，这本书又叫 " 龙书 "，其全面、深入地探讨了编译器设计方面的重要主题，包括词法分析、语法分析、语法制导定义和语法制导翻译、运行时刻环境、目标代码生成、代码优化技术、并行性检测以及过程间分析技术，并在相关章节中给出大量的实例。与上一版相比，本书进行了全面的修订，涵盖了编译器开发方面的最新进展。每章中都提供了大量的系统及参考文献。

小结

好了，最后我们来总结一些今天分享的内容。在这篇文章中，我建议想进入专业编程领域的人，一定要算法、数据结构、网络模型、计算机原理等理论知识，并推荐了相应的学习素材，给出了我的思考和建议。

我认为，虽然这些理论知识枯燥难学，而且通常学完了在工作中也并不是马上就能用上，但这些知识是必须要学好的。这些理论知识可以说是计算机科学这门学科最精华的知识了，认真学习，理解其背后的逻辑和思维方式，会令你受益匪浅。不管是未来学习新知识，还是解决什么疑难问题，都能在这些知识中获得灵感或者启发。

下篇文章中，我们将分享每个程序员都需要掌握的系统知识。敬请期待。

下面是《程序员练级攻略（2018）》系列文章的目录（持续更新中）。

- [开篇词](#)
- 入门篇
 - [零基础启蒙](#)
 - [正式入门](#)
- 修养篇
 - [程序员修养](#)
- 专业基础篇
 - [编程语言](#)
 - [理论学科](#)

- 系统知识

- 软件设计篇
- 高手成长篇



版权归极客邦科技所有，未经许可不得转载

精选留言



服务器

👍 5

还有一本书，很适合打基础，《Computer systems, a programmer's perspective》

2018-06-14



孙悟空

👍 4

深入理解计算机系统 这本书怎么样

2018-06-14

作者回复

很好，下一篇“系统知识”中会推荐。

2018-06-14



Dante.Fung

👍 3

作者是神一般的存在，男神

2018-06-14

作者回复

😊😊😊😊

2018-06-14



繁泽

👍 2

看耗子叔的这系列文章简直是打鸡血，啊啊啊啊我还有这么多这么多书要看！😄

2018-06-14



大P

👍 2

曾今一家伙对我说，你们技术人员就是死脑子，有技术不知道变现.....，看到这篇我想说这是学无止境，真的一头会钻进技术里了，都学完估计人生已过半，不知道对技术人员是不是悲催的事？

2018-06-14



攻城狮

👍 1

算法导论是北美CS本科必学之一

2018-06-14



metalmac.kyle

👍 1

皓哥，想请教下个问题，个人理解您每一篇应该是一个很好的练级参考攻略及引子，那么我在学习的每一步如何考核和量化自己的学习情况呢？比如我如何衡量自己已经完成入门可以进入下一阶段，每一个阶段的针对个人的晋级标准如何定呢？想看看您怎么建议的

2018-06-14



Nowz

👍 0

最近在看sicp，看到一半。lisp作为一门语言来学习是否推荐？

2018-06-15



陈敬秀

👍 0

耗子叔，我偏理论的书怎么看？比如算法导论，后面的习题，该如何处理呢？另外一个就是你描述算法的使用场景，觉得很实用，都想马上尝试，谢谢耗子叔分享

2018-06-14



Phoenix

👍 0

耗子叔，在上上篇程序员修养文章里，您提到，尽量阅读英文教材，是不是可以理解，您上面推荐的书籍，阅读原版会更好？还是说中文版和原版相差不大？

2018-06-14

作者回复

你可以根据自己情况来吧.....

2018-06-15



pomysky

👍 0

毕业时间越长.....越觉得理论基础的重要性

2018-06-14



mingshun

👍 0

非科班生感觉这个书单很重要，毕竟大学错过了。

2018-06-14



oatlmy

0

耗子叔，问一下有关进群审核分享文章的。

我从物联网这个领域切入，介绍整体行业以及我对这个行业的理解，然后再引入相关技术，描述我所做项目中的思考(各方面)，这样可以吗？

我是在读研究生，在老师公司做工业物联网领域开发。因为公司人少是以项目驱动，所以也有幸从硬件、嵌入式再做到软件(现在研二就稳定做web开发)，也接触了一些制造企业，也有了些自己的感悟。

像这样写物联网领域行不行？

2018-06-14



Geek_028aa7

0

基础知识永不落伍。扎实的基础是走向高峰的地基。学习好几年觉得还是井底之蛙。跟着耗哥继续努力

2018-06-14



Leven

0

想问下作者，这些书在什么阶段读比较好，我目前属于在熟悉JAVA基础阶段。

2018-06-14



寻路之人

0

皓叔，可不可以讲一下如何阅读源码，有没有一些心得方法？求回复

2018-06-14



KaiLun

0

耗子叔的专栏订了就是赚到了，总能在你打算休息的时候重新提起劲来。

2018-06-14



Sawyer

0

想问下耗子叔推荐的这些书，你都有阅读过才推荐给大家的吗？因为看到这些书籍的推荐理由好像都是豆瓣上的，如果耗子叔是阅读后推荐的话，那我也可以放心的购买学习了。请耗子叔叔回复一下，谢谢啦。

2018-06-14

作者回复

是的，我都读过的，放心购买。

2018-06-14



dancer

0

要换个书架了☺

2018-06-14



KingPoker

0

越难越要花时间钻研，最重要是和自己方向有关系的，这样才不枯燥

2018-06-14

