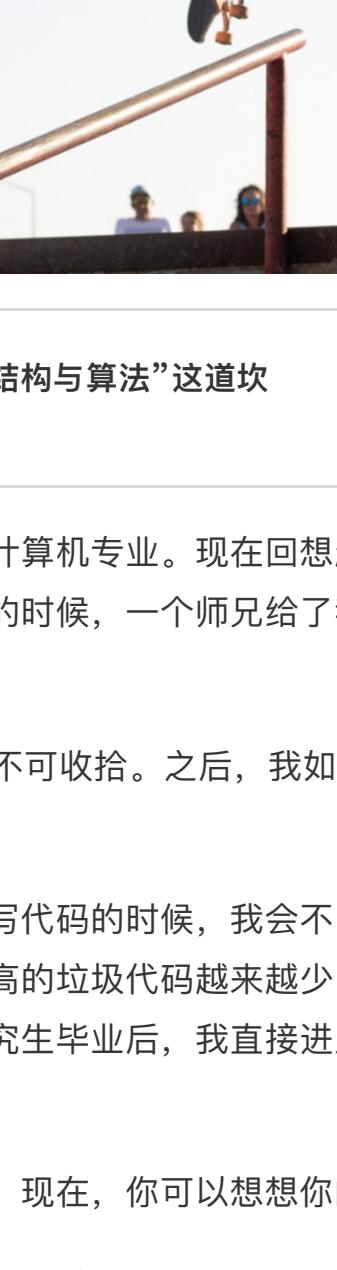


开篇词 | 从今天起，跨过“数据结构与算法”这道坎

2018-09-17 王争



开篇词 | 从今天起，跨过“数据结构与算法”这道坎

朗读人：修阳 08'12'' | 3.76M

你好，我是王争，毕业于西安交通大学计算机专业。现在回想起来，本科毕业的时候，我的编程水平其实是很差的。直到读研究生的时候，一个师兄给了我一本《算法导论》，说你可以看看，对你的编程会很有帮助。

没想到，从此我对算法的“迷恋”便一发不可收拾。之后，我如饥似渴地把图书馆里几乎所有数据结构和算法书籍都读了一遍。

我常常边读边练。没多久，我就发现，写代码的时候，我会不由自主考虑很多性能方面的问题。我写出时间复杂度高、空间复杂度高的垃圾代码越来越少了，算法能力提升了很多，编程能力也有了质的飞跃。得益于此，研究生毕业后，我直接进入 Google，从事 Google 翻译相关的开发工作。

这是我自己学习数据结构与算法的经历，现在，你可以想想你的情况。

- 是不是从学校开始，你就觉得数据结构难学，然后一直没认真学？
- 工作中，一遇到数据结构这个坑，你又发自本能地迅速避让，因为觉得自己不懂，所以也不想深究，反正看起来无关大局？
- 当你想换工作面试，或者研究某个开源项目源码，亦或者和团队讨论某个非框架层面的高可用难题的时候，你又发现，自己的基础跟不上别人的节奏？

如果你是这种情况，其实你并不孤独，这不是你一个人遇到的问题。工作十年间，我见过许多程序员。他们有着各种各样的背景，有很多既有潜力又非常努力，但始终无法在自己现有水平上更进一步。

在技术圈里，我们经常喜欢谈论高大上的架构，比如高可用、微服务、服务治理等等。鲜有人关注代码层面的编程能力，而愿意沉下心来，花几个月时间啃一啃计算机基础知识、认认真真夯实基础的人，简直就是凤毛麟角。

我认识一位原来腾讯 T4 的技术大牛。在区块链大潮之前，他在腾讯工作了 10 多年，长期负责手机 QQ 后台整体建设。他经历了手机 QQ 从诞生到亿级用户在线的整个过程。后来他去了微众银行，有一天老板让他去做区块链。**他用了不到半年时间，就把区块链的整个技术脉络摸清楚了。**现在，他是微众银行的区块链负责人，微众科技创新产品部的老总。你说厉害不？你可以花半年时间就能精通一个新的领域吗？为什么他就可以做到？

我觉得这其中最重要的就是基础足够扎实。他曾经跟我说，像区块链、人工智能这些看似很新的技术，其实一点儿都不“新”。最初学编程的时候，他就把那些基础的知识都学透了。当面临行业变动、新技术更迭的时候，他不断发现，那些所谓的新技术，核心和本质的东西其实就是当初学的那些知识。掌握了这个“规律”之后，他学任何东西都很快，任何新技术都能快速迎头赶上。这就是他快速学习并且获得成功的秘诀。

所以说，**基础知识就像是一座大楼的地基，它决定了我们的技术高度。而要想快速做出点事情，前提条件一定是基础能力过硬，“内功”要到位。**

那技术人究竟都需要修炼哪些“内功”呢？我觉得，无外乎就是大学里的那些基础课程，操作系统、计算机网络、编译原理等等，当然还有数据结构和算法。

可是，我们都知道，像《算法导论》这些经典书籍，虽然很全面，但是过于理论，学起来非常枯燥；而市面很多课程大多缺失真实的开发场景，费劲学完感觉好像还是用不上，过不了几天就忘了。

所以，我尝试做一个**让你能真正受用的数据结构与算法课程，希望给你指明一个简洁、高效的学习路径，教你一个学习基础知识的通用方法**。那么，关于专栏内容，我是怎样设计的呢？

1. 我根据自己研读数十本算法书籍和多年项目开发的经验，在众多的数据结构和算法中，精选了最实用的内容进行讲解。
2. 我不只会教你怎么用，还会告诉你，我们为什么需要这种数据结构和算法，一点点帮你捋清它们背后的设计思想，培养你举一反三的能力。
3. 对于每种数据结构和算法，我都会结合真实的软件开发案例来讲解，让你知道，数据结构和算法，究竟应该如何应用到实际的编码中。

为了由浅入深地带你学习，我把专栏分成四个**递进**的模块。

1. 入门篇
2. 基础篇
3. 高级篇
4. 实战篇

时间、空间复杂度分析是数据结构和算法中非常重要的知识点，贯穿整个专栏的学习过程。但同时也是比较难掌握的，所以我用了 2 节课来讲这部分内容，而且还举了大量的实例，让你一边学一边练，真正能掌握复杂度分析，为后面的学习铺路。

我希望通过这一模块，你能掌握时间、空间复杂度的概念，大 O 表示法的由来，各种复杂度分析技巧，以及最好、最坏、平均、均摊复杂度分析方法。之后，面对任何代码的复杂度分析，你都能游刃有余、毫不畏惧！

1. 入门篇
2. 基础篇
3. 高级篇
4. 实战篇

这部分是专栏中篇幅最大的内容，也是我们学习的重点，共有 26 节内容，涵盖了最基础、最常用的数据结构和算法。针对每种数据结构和算法，我都会结合具体的软件开发实例，由浅入深进行讲解，并适时总结一些实用“宝典”，保证你印象深刻、学有所用。

比如递归这一节，我会讲到，为什么递归代码比较难写？如何避免堆栈溢出？如何避免递归冗余计算？如何将递归代码转化为非递归代码？

我希望，通过这个专栏，不仅能帮你跨过数据结构与算法这个坎，还能帮你掌握一种学习知识和技能的方法，帮你度过职场甚至人生的重要时刻！一起加油吧！