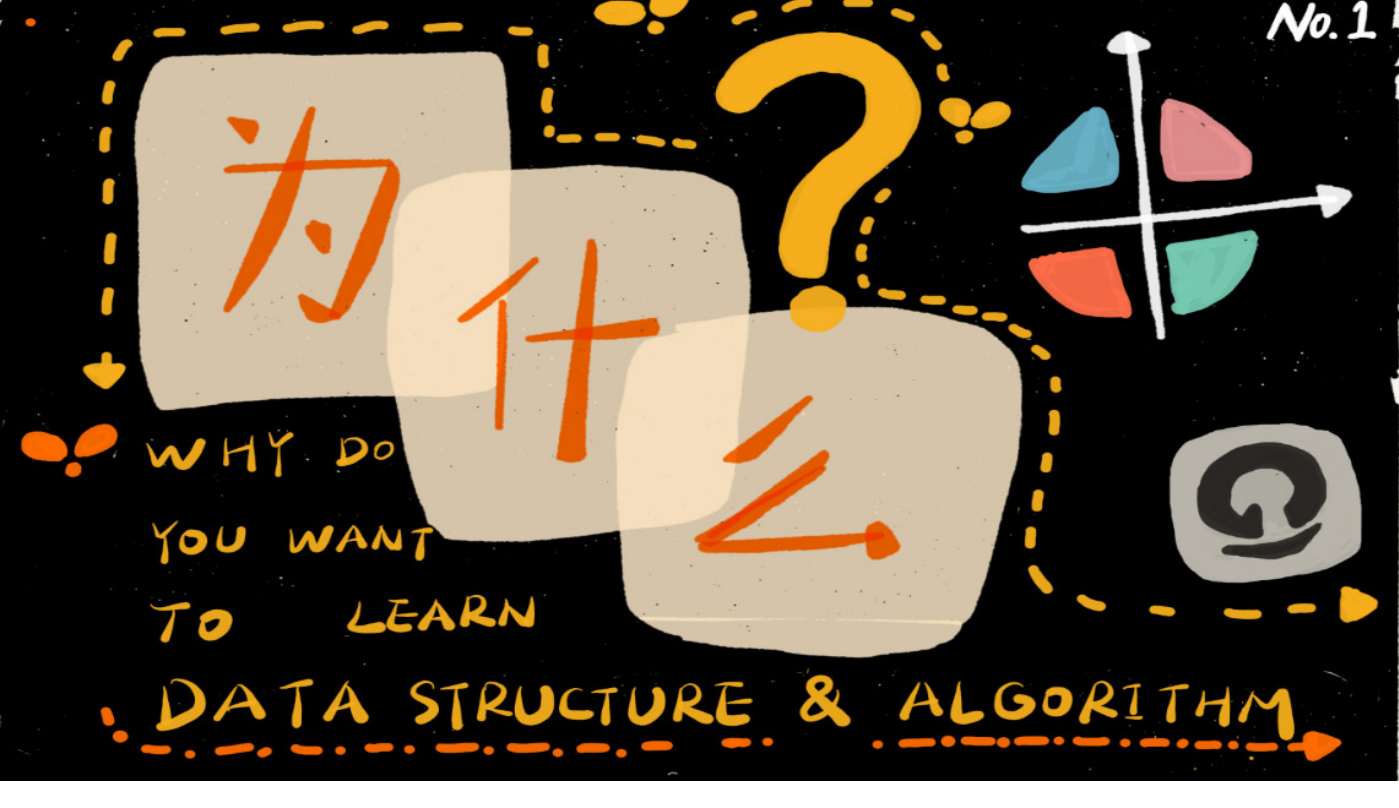


## 01 | 为什么要学习数据结构和算法？

2018-09-21 王争



### 01 | 为什么要学习数据结构和算法？

朗读者：修阳 08'59" | 4.13M

你是不是觉得数据结构和算法，跟操作系统、计算机网络一样，是脱离实际工作的知识？可能除了面试，这辈子也用不着？

尽管计算机相关专业的同学在大学都学过这门课程，甚至很多培训机构也会培训这方面的知识，但是据我了解，很多程序员对数据结构和算法依旧一窍不通。还有一些人也只听说过数组、链表、快排这些最最基本的数据结构和算法，稍微复杂一点的就完全没概念。

当然，也有很多人说，自己实际工作中根本用不到数据结构和算法。所以，就算不懂这块知识，只要 Java API、开发框架用得熟练，照样可以把代码写得“飞”起来。事实真的是这样吗？

今天我们就来详细聊一聊，为什么要学习数据结构和算法。

### 想要通关大厂面试，千万别让数据结构和算法拖了后腿

很多大公司，比如 BAT、Google、Facebook，面试的时候都喜欢考算法、让人现场写代码。有些人虽然技术不错，但每次去面试都会“跪”在算法上，很是可惜。那你有没有想过，为什么这些大公司都喜欢考算法呢？

校招的时候，参加面试的学生通常没有实际项目经验，公司只能考察他们的基础知识是否牢固。社招就更不用说了，越是厉害的公司，越是注重考察数据结构与算法这类基础知识。相比短期能力，他们更看中你的长期潜力。

你可能要说了，我不懂数据结构与算法，照样找到了好工作啊。那我是不是就不用学数据结构和算法呢？当然不是，你别忘了，**我们学任何知识都是为了“用”的，是为了解决实际工作问题的**，学习数据结构和算法自然也不例外。

### 业务开发工程师，你真的愿意做一辈子 CRUD boy 吗？

如果你是一名业务开发工程师，你可能要说，我整天就是做数据库 CRUD（增删改查），哪里用得到数据结构和算法啊？

是的，对于大部分业务开发来说，我们平时可能更多的是利用已经封装好的现成的接口、类库来堆砌、翻译业务逻辑，很少需要自己实现数据结构和算法。但是，**不需要自己实现，并不代表什么都不需要了解**。

如果不知道这些类库背后的原理，不懂得时间、空间复杂度分析，你如何能用好、用对它们？存储某个业务数据的时候，你如何知道应该用 ArrayList，还是 Linked List 呢？调用了某个函数之后，你又如何评估代码的性能和资源的消耗呢？

作为业务开发，我们会用到各种框架、中间件和底层系统，比如 Spring、RPC 框架、消息中间件、Redis 等等。**在这些基础框架中，一般都揉和了很多基础数据结构和算法的设计思想**。

比如，我们常用的 Key-Value 数据库 Redis 中，里面的有序集合是用什么数据结构来实现的呢？为什么要用跳表来实现呢？为什么不用二叉树呢？

如果你能弄明白这些底层原理，你就能更好地使用它们。即便出现问题，也很容易就能定位。因此，**掌握数据结构和算法，不管对于阅读框架源码，还是理解其背后的设计思想，都是非常有用的**。

在平时的工作中，数据结构和算法的应用到处可见。我来举一个你非常熟悉的例子：如何实时地统计业务接口的 99% 响应时间？

你可能最先想到，每次查询时，从小到大排序所有的响应时间，如果总共有 1200 个数据，那第 1188 个数据就是 99% 的响应时间。很显然，每次用这个方法查询的话都要排序，效率是非常低的。但是，如果你知道“堆”这个数据结构，用两个堆可以非常高效地解决这个问题。

### 基础架构研发工程师，写出达到开源水平的框架才是你的目标！

现在互联网上的技术文章、架构分享、开源项目满天飞，照猫画虎做一套基础框架并不难。我就拿 RPC 框架举例。

不同的公司、不同的人做出的 RPC 框架，架构设计思路都差不多，最后实现的功能也都差不多。但是有的人做出来的框架，Bug 很多、性能一般、扩展性也不好，只能在自己公司仅有的几个项目里面用一下。而有的人做的框架可以开源到 GitHub 上给很多人用，甚至被 Apache 收录。为什么会有这么大的差距呢？

我觉得，高手之间的竞争其实就在细节。这些细节包括：你用的算法是不是够优化，数据存取的效率是不是够高，内存是不是够节省等等。这些累积起来，决定了一个框架是不是优秀。所以，如果你还不懂数据结构和算法，没听说过大 O 复杂度分析，不知道怎么分析代码的时间复杂度和空间复杂度，那肯定说不过去了，赶紧来补一补吧！

### 对编程还有追求？不想被行业淘汰？那就不要只会写凑合能用的代码！

何为编程能力强？是代码的可读性好、健壮？还是扩展性好？我觉得没法列，也列不全。但是，在我看来，**性能好坏起码是其中一个非常重要的评判标准**。但是，如果你连代码的时间复杂度、空间复杂度都不知道怎么分析，怎么写出高性能的代码呢？

你可能会说，我在小公司工作，用户量很少，需要处理的数据量也很少，开发中不需要考虑那么多性能的问题，完成功能就可以，用什么数据结构和算法，差别根本不大。但是你真的想“十年如一日”地做一样的工作吗？

经常有人说，程序员 35 岁之后很容易陷入瓶颈，被行业淘汰，我觉得原因其实就在此。有的人写代码的时候，从来都不考虑非功能性的需求，只是完成功能，凑合能用就好；做事情的时候，也从来没有长远规划，只把眼前事情做好就满足了。

我曾经面试过很多大龄候选人，简历能写十几页，经历的项目有几十个，但是细看起来，每个项目都是重复地堆砌业务逻辑而已，完全没有难度递进，看不出有能力提升。久而久之，十年的积累可能跟一年的积累没有任何区别。这样的人，怎么不会被行业淘汰呢？

如果你在一家成熟的公司，或者 BAT 这样的大公司，面对的是千万级甚至亿级的用户，开发的是 TB、PB 级别数据的处理系统。性能几乎是开发过程中时刻都要考虑的问题。一个简单的 ArrayList、Linked List 的选择问题，就可能产生成千上万倍的性能差别。这个时候，数据结构和算法的意义就完全凸显出来了。

其实，我觉得，数据结构和算法这个东西，如果你不去学，可能真的这辈子都用不到，也感受不到它的好。但是一旦掌握，你就会常常被它的强大威力所折服。之前你可能需要费很大劲儿来优化的代码，需要花很多心思来设计的架构，用了数据结构和算法之后，很容易就可以解决了。

### 内容小结

我们学习数据结构和算法，并不是为了死记硬背几个知识点。我们的目的是建立时间复杂度、空间复杂度意识，写出高质量的代码，能够设计基础架构，提升编程技能，训练逻辑思维，积攒人生经验，以此获得工作回报，实现你的价值，完善你的人生。

所以，不管你是业务开发工程师，还是基础架构工程师；不管你是初入职场的初级工程师，还是工作多年的资深架构师，又或者是想转人工智能、区块链这些热门领域的程序员，数据结构与算法作为计算机的基础知识、核心知识，都是必须要掌握的。

**掌握了数据结构与算法，你看待问题的深度，解决问题的角度就会完全不一样**。因为这样的你，就像是站在巨人的肩膀上，拿着生存利器行走世界。数据结构与算法，会为你的编程之路，甚至人生之路打开一扇通往新世界的大门。

### 课后思考

你为什么要学习数据结构和算法呢？在过去的软件开发中，数据结构和算法在哪些地方帮到了你？

欢迎留言和我分享，我会第一时间给你反馈。