

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，你又该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，但是细看下来，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，就可能会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。

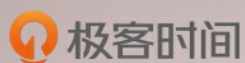
所以，不管你是业务开发工程师，还是基础架构工程师；不管你是初入职场的初级工程师，还是工作多年的资深架构师，又或者者是想转人工智能、区块链这些热门领域的程序员，数据结构与算法作为计算机的基础知识、核心知识，都是必须要掌握的。

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。



数据结构与算法之美

为工程师量身打造的数据结构与算法私教课

王争

前 Google 工程师



新版升级：点击「👤 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

精选留言



姜威

为什么学习数据结构和算法？我认为有3点比较重要

- 1.直接好处是能够有写出性能更优的代码。
- 2.算法，是一种解决问题的思路和方法，有机会应用到生活和事业的其他方面。
- 3.长期来看，大脑思考能力是个人最重要的核心竞争力，而算法是为数不多的能够有效训练大脑思考能力的途径之一。

2018-09-21 06:49

作者回复

写的很好 同学们把这条回复顶上去

2018-09-21 10:07



HouShangLing

一定要动手写

2018-09-21 06:41

作者回复

你掌握了学这门课的最有效的方法。看十遍也没自己实现一遍学的牢。同学们这条也帮忙顶上去

2018-09-21 10:08



wistbean

其实问题的所在就是现在有很多现成的框架，器又大活又好，拿来就用，还不用太担心性能的问题。就好像那些建楼的工程师不需要懂砖瓦的构成，也能盖楼。司机不需要懂汽车引擎的原理，也能当顺风车司机载美女兜风。

遇到不会的上 Google，懒了就上 GitHub 找框架。所以写了这么多年代码，一直是个菜鸟。

或许是时候修炼自己的内功了。一直 CURD 有意思么？把设计模式，网络原理，数据结构和算法捡起来，或许就没那么容易菜鸟了。怼人也更加自信了。跳槽也不用畏畏缩缩的了。

为了不当菜鸟，我和我的小伙伴们就加入了哇！

2018-09-21 03:38

作者回复

写的太好了

2018-09-21 10:12



两颗小虎牙

可不可以每次多发布几篇？

2018-09-21 00:14



tdytaylor

老师，就像你说的，工作中其实好多都和业务挂钩，基本上都是针对业务做增删改查，很难把所学的算法应用起来，我平时也时不时学习算法，但总感觉学了就忘，忘了又学，如此反复，老师，这种到底是没了解算法的原理导致不会灵活应用，还是写的少导致的，感觉学习算法很少能应用起来

2018-09-21 00:26

作者回复

1. 客观的讲，有些项目确实涉及的数据结构和算法少一些，你可以再看下我文章里写的。
2. 你提到学了又忘，我觉得一方面你是没有掌握学习的方法，学习的重点，走马观花的看肯定比较容易忘；我们02节会具体讲；
3. 不会灵活应用？那估计还是没有好的教材教你如何应用，还有可能就是确实还没掌握太牢，只是懂点皮毛，很浅，灵活应用是一个比较的境界，需要一段时间的沉淀学习。
4. 学习算法并不是为了记住几个排序、二分查找、二叉树遍历，他还能锻炼你的逻辑思维、性能意识，而且，如果你写代码能力还有欠缺，你还可以通过把学到的数据结构和算法都实现一遍，这是一种很好很好的锻炼编程能力的方法。所以不要过度追求一定要在项目里手写快排、手写二叉树才能算是用上。

2018-09-21 11:33



裴海港

年近40的大叔也有一颗积极向上的心

2018-09-21 01:29

作者回复

终身学习 多大年纪也不晚

2018-09-21 10:17



五岳寻仙

老师好！看到专栏第一眼就果断订阅了。我是一个菜鸟程序员，半年的工作经验让我感受到数据结构和算法太重要了！讲一个自己亲身经历的例子。

入职不久，就遇到一个需求，需要建立一个3G(30亿)条键值对映射，已供后续检索。听上去很简单的问题，用python的字典就可以解决。但在实现的过程中，很快就遇到了问题：字典是基于hash的，对于每条键值对要多消耗50个字节的内存维持这种结

构(即便使用redis也需要这个内存花销), 再加上键值存储消耗的内存, 我大约需要 $3G \times 70 = 210G$ 内存, 超过服务器内存了。后来, 想到了可以根据键排序后线性存储, 使用二分查找, 解决了这个问题, 大约消耗的内存也就 $3G \times 10 = 30G$ 左右。看似简单的问题, 当规模大到一定程度, 不借助算法和数据结构, 就无法解决了。

2018-09-21 06:55

作者回复

哈哈 你要是看到我讲的散列那一篇你就知道了 像java里的hashmap是比较耗内存的 你用到的解决方案是一种用时间复杂度换空间复杂度的思路 我们专栏也会讲的 不过你现在的解决办法还可以更高效 利用hash函数 我们专栏也会讲到 还有二分是 $\log n$ 的时间复杂度 是非常高效的一种时间复杂度 2的64次方个有序数据二分查找也顶多循环64次 有没有觉得 $\log n$ 这个复杂度很奇妙

2018-09-21 12:29



Haoz

一、数据结构和算法是什么

- 1、数据结构是指一组数据的存储结构
- 2、算法就是操作数据的方法
- 3、数据结构和算法是相辅相成的, 数据结构是为算法服务的, 而算法要作用在特定的数据结构之上

二、学习的重点在什么地方

数据结构和算法解决的是如何更省、更快地存储和处理数据的问题, 因此, 我们就需要一个考量效率和资源消耗的方法, 这就是复杂度分析方法。在学习数据结构和算法的过程中, 要学习它的「来历」、「自身的特点」、「适合解决的问题」以及「实际的应用场景」。

- 1、数据结构和算法学习的精髓-复杂度分析
- 2、最常用的、最基础的数据结构: 数组、链表、栈、队列、散列表、二叉树、堆、跳表、图、Trie 树
- 3、最常用的算法: 递归、排序、二分查找、搜索、哈希算法、贪心算法、分治算法、回溯算法、动态规划、字符串匹配算法

感谢老师的分析, 一直以来数据结构和算法都是我的硬伤, 学了很多次, 也放弃了很多次, 可能是方法不对, 但更多的是因为自己之前的毅力不够。

在此立下 flag: 从本篇文章开始, 将老师的文章根据自己的理解进行输出相应的总结和思考, 发表成留言, 并将每周三篇文章中涉及的数据结构和算法用 Java 实现一次。

2018-09-25 22:37

作者回复

写的很好

2018-09-25 22:44



钳子钳子千

大二上, 刚刚学完第二章顺序表和链表 (一切为了性能xd (半夜等更居然真的刷出来了 (x

2018-09-21 00:12



小麦2018

我还是觉得王争老师自己读的音频好, 亲切, 有听作者亲自讲解的感觉。

2018-09-21 09:00



邵峰

记得以前看金庸小说, 一个人练降龙十八掌, 一个人练全真教内功, 刚开始练内功的被吊打, 十年后练内功的依然被吊打, 但是20年后, 降龙十八掌就怎么也打不过练内功的了, 算法和数据结构就是内功, 降龙十八掌我觉得就是各种框架, 再怎么熟悉框架, 也不如了解它内部的算法和数据结构

2018-09-21 13:33

作者回复

太形象了:)

2018-09-21 15:50



DDT

就是想升工资吧 我知道学技术必须要拿出十年磨一剑的精神 从头开始...

2018-09-21 08:22

作者回复

十年磨一剑说的太好了。我也是这么认为的。做技术就是不要浮躁。要耐得住寂寞。沉得下心。

2018-09-21 09:39



Liar

不做一个没有追求的码农，那些说过了35就不行的码农大部分都是不求上进的人。

2018-09-21 11:27

作者回复

说得好 不管哪个行业都会淘汰不求上进的人 it也不例外

2018-09-21 11:35



阿康

老师，我是一名iOS开发，怎么听完你这篇文章还是觉得算法对于我开发来说不是那么重要 后面我有点怕自己听不懂，而且动手写，我该如何下笔，感觉自己很迷茫.....

2018-09-21 15:47

作者回复

1. 假设我们现在要做这样一个功能，我们希望在app上存储一个多级地址列表，用户可以一层一层的选择地址列表中的地址，来设置自己的所在的省份、市、区，如果这个地址列表不是经常变动的，我们希望保存在app端，这样就不需要每次操作地址列表都要跟后台交互，如果让你存储这个多级地址列表，你会怎么存储呢？

2. 如果地址列表也并不是一直都不变的，如果地址列表改变了，我们又不希望发新版，那如何更新app上的地址列表呢？如果地址列表比较大，我们不希望app重新全量的从服务器再拉一次，那又如何来做呢？

3. 我会用一种语言来实现 你可以翻译成object c

2018-09-21 16:48



小行家

为什么学习数据结构与算法？

1. 作为一名程序员，我觉得这个是基本功，就好比建房子一样，地基一定要牢固

2. 现在有如此多的开源项目，人们往往开发中只是照搬，调试，集成，然后很少会去注意性能和优化方面，慢慢的也就成了CRUD boy

3. 算法提供的是一种思想，这个思想真的可以让你超神，让你对待问题有着独特的见解，我觉得这个才是最重要的。不再是原来的，没有思考

4. 数据结构作为cs的基本专业，并且贯穿整个cs学习生涯，在cs方面的研究还是有很大的实用价值。

数据结构与算法的意义？

1. 在本科开发中，曾经就自己写了一个类似于目录搜索的算法，就是简单的for循环，性能极其低下，改为二叉树后，性能大大提升

2. 目前读研，接触机器学习理论和深度学习，原有的算法和数据结构的的学习中，你所掌握的思想，确实会有很大的意义。

最后，我认为，学习最重要的是你能够自己成长，这个才是最棒的。

2018-09-21 12:59

作者回复

说的好

2018-09-21 15:52



lane

现实，都是面相领导编程.....一言难尽...

2018-09-21 01:46

作者回复

没事的 如果工作不满意 不顺心 更要卧薪尝胆 提高能力 沉淀自己。总有一天 机会会到来 你做好充分的抓住它的准备就好。人在职场中 只要抓住一两次大的机会 就能做到很高的职位 就怕的是机会来了我们也抓不住

2018-09-21 10:17



grandcool

没理解统计业务接口99%响应时间啥意思

2018-09-21 08:06

作者回复

举一个例子 你写了一个接口 每天有成千上万的访问 你如何知道这个接口够不够快？响应时间是1s还是5s？如何统计度量？用平均值？显然不是太适合？那用什么值来统计度量呢？你可以自己搜索研究下

2018-09-21 09:53



SelfImpr李

三年小开发来说一波：上周生产系统上遇到了Java的内存溢出情况（可以买彩票了，OOM事件一直在我们的书上），经过分析dump文件，发现Java的类和Java的动态语言生成的对象没有及时释放，我尽全力就修改了代码上，该释放释放；再过了一个星期分析dump文件，发现少了很多，但是还有一些遗留着，我想是时候需要《数据结构和算法》来帮我了！PS：之前也看过红壳得《算法导论》，看了一周，然后又用来垫显示器了，来来回回几次，最终还是垫显示器了。

2018-09-21 10:57

作者回复

oom这种问题看似很遥远 实际上我还真遇到过很多次 多数是从数据库捞数据时 一不小心捞了太多的数据到内存 就oom了 而且这种情况用mat等工具也无法分析 因为这种oom是一过性的 heap dump的时候可能数据已经不可达 而mat这些工具只能分析可达对象

2018-09-21 11:55



Pardo

大学的时候没意识，工作之后才发现，不熟悉数据结构和算法，写出的代码虽然能用，但完全经不起推敲。随着项目复杂度的提高，没有算法护体，代码基本上会是一塌糊涂，迷失在细节调整和障碍修复中不能自拔。

2018-09-21 10:26



Mortimey

作为一枚iOS开发者，感觉里面好多的专业术语是我没听过的，请问老师，后期的课程也是这样的吗？

2018-09-21 08:25

作者回复

嗯嗯 后面会好点

2018-09-21 09:36