每个工程师都应该了解的：大数据时代的算法

2017-12-01 朱赟

每个工程师都应该了解的：大数据时代的算法
朱赟
- 00:17 / 09:50

开了这个专栏之后，经常有读者问算法相关的问题。

> 能不能讲讲算法在工作中的运用？你个人学习算法的过程是怎样的？我对算法还是有点怕。除此之外，你认为大学是应该多花时间学应用技术还是理论知识呢？谢谢。

今天就来聊聊我自己学习算法的过程，以及算法在实际工作中的应用。

以前，我们认为大数据总是优于好算法。也就是说，只要数据量足够大，即使算法没有那么好，也会产生好的结果。

前一阵子“极客时间” App 发布了一条极客新闻：“算法比数据更重要，AlphaGo Zero 完胜旧版。”新闻的内容是谷歌人工智能团队 DeepMind 发布了新版的 AlphaGo 计算机程序，名为 AlphaGo Zero。这款软件能够从空白状态开始，不需要人类输入任何命令，便可以迅速自学围棋，并以 100 比 0 的战绩击败了上一代 AlphaGo。

AlphaGo Zero 最大的突破在于实现了“白板理论”。白板理论认为：婴儿是一块白板，可以通过后天学习和训练来提高智力。AI的先驱图灵认为，只要能用机器制造一个类似于小孩的AI，然后加以训练，就能得到一个近似成人智力，甚至超越人类智力的 AI。

自学成才的 AlphaGo Zero 正是实现了这一理论。AlphaGo 的首席研究员大卫·席尔瓦（David Silver）认为，从 AlphaGo Zero 中可以发现，算法比所谓的计算或数据量更为重要。事实上，AlphaGo Zero 使用的计算要比过去的版本少一个数量级，但是因为使用了更多原理和算法，它的性能反而更加强大。

由此可见，在大数据时代，算法的重要性日渐明晰。一个合格的程序员，必须掌握算法。

我不知道大家是怎样一步步开始精通算法和数据结构的。大二时，我第一次接触到了《数据结构》，因为从来没有过这方面的思维训练，当时的我学习这门课比较费力。那时候接触到的编程比较少，所以并没有很多实际经验让我欣赏和体味：一个好的数据结构和算法设计到底“美”在哪里。

开始学习的时候，我甚至有点死记硬背的感觉，我并不知道“如果不这样设计”，实际上会出现哪些问题。各种时间和空间复杂度对我而言，也仅仅是一些不能融入到实际问题的数学游戏。至于“每种最坏情况、平均情况的时间空间复杂度与各种排序”，这些内容为什么那么重要，当时我想，可能因为考试会考吧。

没想到后来的时日，我又与算法重新结缘。可能是因为莱斯大学给的奖学金太高了，所以每个研究生需要无偿当五个学期的助教 。好巧不巧，我又被算法老师两次挑中当助教。所以，在命运强制下，一本《算法导论》就这样被我前前后后仔细学习了不下四遍。这样的结果是，我基本做过整本书的习题，有些还不止做了一遍。我学习算法的过程，就是反复阅读《算法导论》的过程。

那么，学习算法到底有什么用处呢？

首先，算法是面试的敲门砖

国内的情况我不太清楚，但就硅谷的 IT 公司而言，不但电话面试偏算法，现场面试至少有两轮都是考算法和编程的。

大一些老一些的公司，像谷歌、Facebook、领英、Dropbox等，都是直接在白板上写程序。小一些新一些的公司，如 Square、Airbnb等，都是需要现场上机写出可运行的程序。Twitter、Uber 等公司则是白板上机兼备，视情况而定。

虽说还有其它考系统设计等部分，但如果算法没有打好基础，第一关就很难过，而且算法要熟悉到能够现场短时间内写出正解，所以很多人准备面试前都需要刷题。

有一次我当面试官，电话面试另外一个人，当时是用 Codepad 共享的方式，让对方写一个可运行的正则表达式解析器。45 分钟过去了，对方并没有写出来。我就例行公事地问：“你还有什么问题想问或者想了解么？”对方估计因为写不出程序很有挫败感，就反问：“你们平时工作难道就是天天写正则表达式的解析器么？”

一瞬间，我竟无言以对。想了想，我回复说：“不用天天写。那我再给你 15 分钟，你证明给我看你还会什么，或者有什么理由让我给你进一步面试的机会？”对方想了一会，默默挂掉了电话。

老实说，我对目前面试中偏重算法的程度是持保留意见的。算法题答得好，并不能说明你有多牛。牛人也有因为不愿刷题而马失前蹄的时候。但是除了算法测试，显然也没有更好的方法佐证候选人的实力；然而怎样才能最优化面试流程，这也是个讨论起来没完的话题，并且每次讨论必定无果而终。

其次，☒编程时用到的更多是算法思想，而不是写具体的算法

说到实际工作中真正需要使用算法的机会，让我想一想 —— 这个范围应该在 10% 的附近游走。

有些朋友在工作中遇到算法场景多些，有的少些。更多的时候，是对业务逻辑的理解，对程序语言各种特性的熟练使用，对代码风格和模式的把握，各种同步异步的处理，包括代码测试、系统部署是否正规化等等。需要设计甚至实现一个算法的机会确实很少，即使用到，现学可能都来得及。

但是熟悉基本算法的好处在于：如果工作需要读的一段代码中包含一些基本算法思想，你会比不懂算法的人理解代码含义更快。读到一段烂代码，你知道为什么烂，烂在哪，怎么去优化。

当真的需要在程序中设计算法的时候，熟悉算法的你会给出一个更为完备的方案，对程序中出现的算法或比较复杂的时间复杂度问题你会更有敏感性。熟悉算法你还可以成为一个更优秀的面试官，可以和别的工程师聊天时候不被鄙视。

最后，不精通算法的工程师永远不是好工程师

当然，除了算法导论中那些已成为经典的基本算法以及算法思想（Divide-and-conquer，Dynamic programming）等，其实我们每天接触到的各种技术中，算法无处不在。

就拿人人都会接触的存储为例吧，各种不同的数据库或者键值存储的实现，就会涉及各种分片（Sharding）算法、缓存失败（Cache Invalidation）算法、 锁定（Locking）算法，包括各种容错算法（多复制的同步算法）。 虽然说平时不太会去写这些算法 —— 除非你恰恰是做数据库实现的 —— 但是真正做到了解这项技术的算法细节和实现细节，无论对于技术选型还是对自己程序的整体性能评估都是至关重要的。

举个例子，当你在系统里需要一个键值存储方案的时候，面对可供选择的各种备选方案，到底应该选择哪一种呢？

永远没有一种方案在所有方面都是最佳的。就拿 Facebook 开源的 RocksDB 来说吧。了解它历史的人都知道，RocksDB 是构建在 LevelDB 之上的，可以在多 CPU服务器上高效运行的一种键值存储。而 LevelDB 又是基于谷歌的 BigTable 数据库系统概念设计的。

早在 2004 年，谷歌开始开发 BigTable，其代码大量的依赖谷歌内部的代码库，虽然 BigTable 很牛，却因此无法开源。2011 年，谷歌的杰夫·迪恩和桑杰·格玛沃尔特开始基于 BigTable 的思想，重新开发一个开源的类似系统，并保证做到不用任何谷歌的代码库，于是就有了 LevelDB。这样一个键值存储的实现也用在了谷歌浏览器的 IndexedDB 中，对于谷歌浏览器的开源也提供了一定的支持。

我曾经在文章中提到过 CockroachDB，其实又可以看作是基于 RocksDB 之上的一个分布式实现。从另一个层面上讲，CockroachDB 又可以说是 Spanner 的一个开源实现。知道这些，就知道这些数据库或键值存储其实都同出一系。再来看看 LevelDB 底层的 SSTable 算法，就知道他们都是针对高吞吐量（high throughput），顺序读/写工作负载（sequential read/write workloads）有效的存储系统。

当然，一个系统里除了最基本的算法，很多的实现细节和系统架构都会对性能及应用有很大的影响。然而，对算法本身的理解和把握，永远是深入了解系统不可或缺的一环。

类似的例子还有很多，比如日志分析、打车软件的调度算法。

拿我比较熟悉的支付领域来说吧，比如信用卡BIN参数的压缩，从服务端到移动 App 的数据传输，为了让传输数据足够小，需要对数据进行压缩编码。

每个国家，比如中国、韩国、墨西哥信用卡前缀格式都不一样，如何尽量压缩同时又不会太复杂，以至于影响移动 App 端的代码复杂度，甚至形成 Bug 等，也需要对各种相关算法有详尽地了解，才有可能做出最优的方案。

关于算法我们来总结一下：

1. 在大数据时代，数据和算法都同等重要，甚至算法比计算能力或数据量更为重要。
2. 如何学习算法呢？读经典著作、做题，然后在实践中阅读和使用算法。
3. 算法是面试的敲门砖，可以帮助你得到一份自己喜欢的工作。
4. 写程序中用到的更多是算法思想，不是写具体的算法。
5. 不精通算法的工程师永远不会是一个优秀的工程师，只有对各种相关算法有详尽理解，才有可能做出最优的方案。

希望每个读者都成为合格甚至优秀的软件工程师，如果你在工作中遇到过有趣的算法故事，也可以在留言中告诉我。

参考外链：

https://www.igvita.com/2012/02/06/sstable-and-log-structured-storage-leveldb/

蝈蝈

2017-12-01

每个工程师应该要保持打破沙锅问到底的心态，这样才有利于自身知识的沉淀，最终转化为价值提升。在开源框架层出不穷的时代，理应做到对其内在本质（数据结构设计和算法）的理解，以

不变应万变，方能做为知识的主人。

ibrothergang

2017-12-01

现在日常工作中，看得见的地方涉及算法的真的不多，但是也知道真正的提高就在这些看不见的地方。

记得最早真正接触算法是在做一个在线的编程题库，里面的题目从简单到复杂，你需要在线提交你写的程序代码，然后网站会运行你写的代码，主要会从结果的正确性以及运行的性能两方面来评价你写的代码好不好，越到后面，对程序计算花费的时间要求越来越高，导致对算法以及数据结构的要求也相应的越来越高。那时候成功通过一道编程题时的那种喜悦，那种兴奋，应该只有经历过的人才懂。

算法，在学习，了解，熟悉，掌握的过程中，你会体会和一般编程完全不一样的感受！

clpsz

2017-12-01

有些设计很巧妙的算法虽然工作中未必会用上，但对于思维方式的影响还是很大的

咸鱼

2017-12-07

你好，请问对于你刚开始接触算法的那段感受和经历的描写，是百分百真实的，还是考虑到读者的大众水平而加工修饰后的？
我16年本科毕业，目前在中国二线互联网企业工作，遇到专业能力的瓶颈，我很怀疑自己的潜力，看到这篇文章，我觉得我学习方式有问题，我喜欢只做了解，很少深入。
谢谢。

zhengfc

2017-12-01

不知道算法和可读性有时候是不是有点冲突

野山门

2017-12-01

用算法来解决实际问题的过程是最有趣的。大学毕业的时候自己从头到尾实现了一个台球游戏，里面用到了物理动能守恒、摩擦系数等等。非常有趣，我能想象使用算法解决现实生活中其他问题的时候也同样有趣。

William李梓峰

2017-12-01

谢谢安姐指点，我可以毫无顾忌地复习算法了。

simaopig

2017-12-01

开始关注算法，从现在就开始

MarksGui

2018-05-21

不知道中文版的《算法》第四版做为入门的学习教材行吗？有没有什么好的推荐学习方法了？

Desperado

2018-05-02

国内业务程序员远占比太大，需要用到算法的地方很少很少

Dylan

2017-12-28

很惭愧，毕业这么多年《算法导论》这本书还没读完一遍～对于算法个人还是很喜欢的，对动态规划还有贪心算法一直记忆犹新～不管是曾经面试其它公司还是现在面试工程师算法题一定是要问的,因为就像作者说的，初面考察工程师解决问题的能力除了算法题我不知道还有没有更好的方法，业务能力或者其它能力可以后续培养，但是我相信分析和解决问题的能力是很难去培养的

秋水天

2017-12-15

大公司对算法要求确实很高。

幻想

2017-12-07

这篇文章太有价值了，多谢安姐姐

shniu

2017-12-07

在我深入研究算法的道路上又多了一份坚信^_^

bluze

2017-12-05

好吧 你说服我继续研究算法啦

Seven _ dong

2017-12-03

安姐具体讲讲怎么学习的?

Silence Wang

2017-12-02

对于大部分软件设计师，算法更多的是在锻炼人的逻辑思维能力。