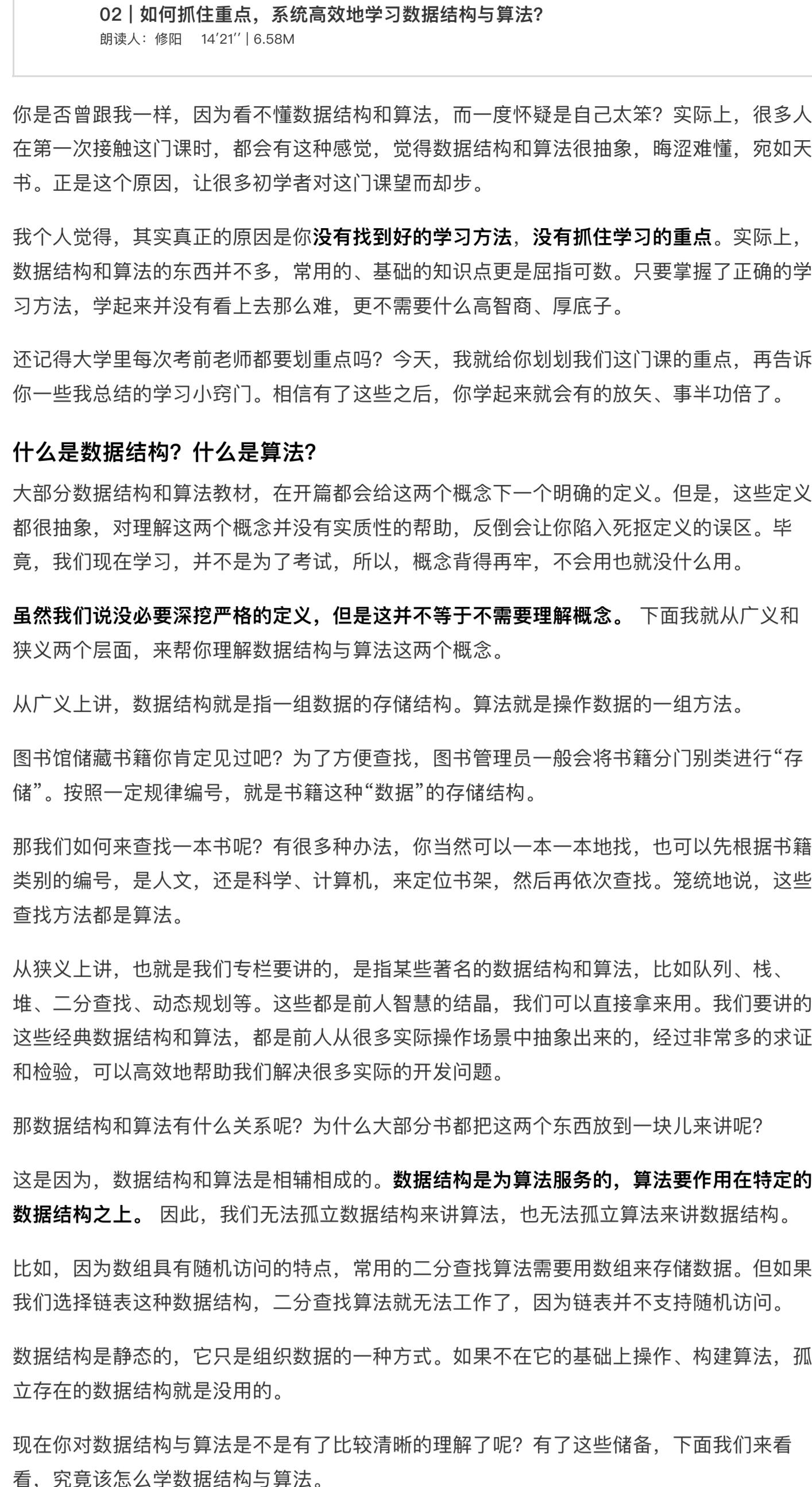


02 | 如何抓住重点，系统高效地学习数据结构与算法？

2018-09-24 王争



02 | 如何抓住重点，系统高效地学习数据结构与算法？

朗读人：修阳 14'21" | 6.58M

你是否曾跟我一样，因为看不懂数据结构和算法，而一度怀疑是自己太笨？实际上，很多人在第一次接触这门课时，都会有这种感觉，觉得数据结构和算法很抽象，晦涩难懂，宛如天书。正是这个原因，让很多初学者对这门课望而却步。

我个人觉得，其实真正的原因是你没有找到好的学习方法，没有抓住学习的重点。实际上，数据结构和算法的东西并不多，常用的基础的知识点更是屈指可数。只要掌握了正确的学习方法，学起来并没有看上去那么难，更不需要什么高智商、厚底子。

还记得大学里每次考前老师都要划重点吗？今天，我就给你划划我们这门课的重点，再告诉你一些我总结的学习小窍门。相信有了这些之后，你学起来就会有的放矢、事半功倍了。

什么是数据结构？什么是算法？

大部分数据结构和算法教材，在开篇都会给这两个概念下一个明确的定义。但是，这些定义都很抽象，对理解这两个概念并没有实质性的帮助，反倒会让你陷入死抠定义的误区。毕竟，我们现在学习，并不是为了考试，所以，概念背得再牢，不会用也就没什么用。

虽然我们说没必要深挖严格的定义，但是这并不等于不需要理解概念。下面我就从广义和狭义两个层面，来帮你理解数据结构与算法这两个概念。

从广义上讲，数据结构就是指一组数据的存储结构。算法就是操作数据的一组方法。

图书馆储藏书籍你肯定见过吧？为了方便查找，图书管理员一般会将书籍分门别类进行“存储”。按照一定规律编号，就是书籍这种“数据”的存储结构。

那我们如何来查找一本书呢？有很多种办法，你当然可以一本一本地找，也可以先根据书籍类别的编号，是人文，还是科学、计算机，来定位书架，然后再依次查找。笼统地说，这些查找方法都是算法。

从狭义上讲，也就是我们专栏要讲的，是指某些著名的数据结构和算法，比如队列、栈、堆、二分查找、动态规划等。这些都是前人智慧的结晶，我们可以直接拿来用。我们要讲的这些经典数据结构和算法，都是前人从很多实际操作场景中抽象出来的，经过非常多的求证和检验，可以高效地帮助我们解决很多实际的开发问题。

那数据结构和算法有什么关系呢？为什么大部分书都把这两个东西放到一块儿来讲呢？

这是因为，数据结构和算法是相辅相成的。数据结构是为算法服务的，算法要作用在特定的数据结构之上。因此，我们无法孤立数据结构来讲算法，也无法孤立算法来讲数据结构。

比如，因为数组具有随机访问的特点，常用的二分查找算法需要用到数组来存储数据。但如果我们选择链表这种数据结构，二分查找算法就无法工作了，因为链表并不支持随机访问。

数据结构是静态的，它只是组织数据的一种方式。如果不在于它的基础上操作、构建算法，孤立存在的数据结构就是没用的。

现在你对数据结构与算法是不是有了比较清晰的理解了呢？有了这些储备，下面我们来看看，究竟该怎么学数据结构与算法。

学习这个专栏需要什么基础？

看到数据结构和算法里的“算法”两个字，很多人就会联想到“数学”，觉得算法会涉及到很多深奥的数学知识。那我数学基础不是很好，学起来会不会很吃力啊？

数据结构和算法课程确实会涉及一些数学方面的推理、证明，尤其是在分析某个算法的时间、空间复杂度的时候，但是这个你完全不需要担心。

这个专栏不会像《算法导论》那样，里面有非常复杂的数学证明和推理。我会由浅入深，从概念到应用，一点一点给你解释清楚。你只要有高中数学水平，就完全可以学习。

当然，我希望你最好有些编程基础，如果有项目经验就更好了。这样我给你讲数据结构和算法如何提高效率、如何节省存储空间，你就会有很直观的感受。因为，对于每个概念和实现过程，我都会从实际场景出发，不仅教你“是什么”，还会教你“为什么”，并且告诉你遇到同类型问题应该“怎么做”。

学习的重点在什么地方？

提到数据结构和算法，很多人就很头疼，因为这里面的内容实在是太多了。这里，我就帮你梳理一下，应该先学什么，后学什么。你可以对照看看，你属于哪个阶段，然后有针对性地进行学习。

想要学习数据结构与算法，首先要掌握一个数据结构与算法中最重要的概念——复杂度分析。

这个概念究竟有多重要呢？可以这么说，它几乎占了数据结构和算法这门课的半壁江山，是数据结构和算法学习的精髓。

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。所以，如果你只掌握了数据结构和算法的特点、用法，但是没有学会复杂度分析，那就相当于只知道操作口诀，而没掌握心法。只有把心法了然于胸，才能做到无招胜有招！

所以，复杂度分析这个内容，我会用很大篇幅给你讲透。你也一定要花大力气来啃，必须要拿下，并且要搞得非常熟练。否则，后面的数据结构和算法也很难学好。

搞定复杂度分析，下面就要进入数据结构与算法的正文内容了。

为了让你对数据结构和算法能有个全面的认识，我画了一张图，里面几乎涵盖了所有数据结构和算法书籍中都会讲到的知识点。

（图谱内容较多，建议长按保存后浏览）

但是，作为初学者，或者一个非算法工程师来说，你并不需要掌握图里面的所有知识点。很多高级的数据结构与算法，比如二分图、最大流等，这些在我们平常的开发中很少会用到。所以，你暂时可以不用看。我还是那句话，咱们学习要学会找重点。如果不分重点地学习，眉毛胡子一把抓，学起来肯定比较吃力。

所以，结合我自己的学习心得，还有这些年的面试、开发经验，我总结了20个最常用、最基础数据结构与算法，不管是应付面试还是工作需要，只要集中精力逐一攻克这20个知识点就足够了。

这里面有10个数据结构：数组、链表、栈、队列、散列表、二叉树、堆、跳表、图、Trie树。这里有10个算法：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。

掌握了这些基础的数据结构和算法，再学更加复杂的数据结构和算法，就会非常容易、非常快。

在学习数据结构和算法的过程中，你也要注意，不要只是死记硬背，不要为了学习而学习，而是要学习它的“来历”“自身的特点”“适合解决的问题”以及“实际的应用场景”。对于每种数据结构或算法，我都会从这几个方面进行详细讲解。只要你掌握了我每节课里讲的内容，就能在开发中灵活应用。

从我个人的观点是可以“适度”刷题，但一定不要浪费太多时间在刷题上。我们学习的目的还是掌握，然后应用。除非你要面试 Google、Facebook 这样的公司，它们的算法题目非常难，必须大量刷题，才能在短期内提升应试正确率。如果是应对国内公司的技术面试，即便是 BAT 这样的公司，你只要彻底掌握这个专栏的内容，就足以应对。

所以，我们在初学的过程中，也可以给自己设立一个切实可行的目标，就像打怪升级一样。

比如，针对这个专栏，你就可以设立这样一个目标：每节课后的思考题都认真思考，并且回复到留言区。当你看到很多人给你点赞之后，你就会有成就感。因为，这其实帮你改掉了一点学习的习惯。

所以，我们初学的过程中，也可以给自己设立一个切实可行的目标，就像打怪升级一样。

当然，还有很多其他的目標，比如，每节课后都写一篇学习笔记或者心得；或者你还可以每节课都找一下我讲得不对、不合理的地方……诸如此类，你可以总结一个适合你的“打怪升级攻略”。

如果你能这样学习一段时间，不仅能够收获到知识，你还会有意想不到的成就感。因为，这其实帮你改掉了一点学习的习惯。

4. 知识需要沉淀，不要想试图一下子掌握所有

在学习的过程中，一定会碰到“拦路虎”。如果哪个知识点没有怎么学懂，不要着急，这是正常的。因为，想听一遍，看一遍就把所有知识掌握，这肯定是不可能的。学习知识的过程是反复迭代、不断沉淀的过程。

如果碰到“拦路虎”，你可以尽情地在留言区问我，也可以先沉淀一下，过几天再重新学习一遍。所谓“读书百遍，其义自见”，我觉得这是很有道理的！

我讲的这些学习方法，不仅仅针对咱们这一个专栏，其实完全适用于任何知识的学习过程。你可以通过这个专栏，实践一下这些方法。如果效果不错，再推广到之后的学习过程中。

另外，你也可以把划数据结构和算法的学习重点，复杂度分析，以及10个数据结构和10个算法。

这些内容是我们根据平时的学习和工作、面试经验积累，精心挑选出来的。只要掌握了这些内容，应付日常的面试、工作，基本不会有问题。

除此之外，我还给你分享了我总结的一些学习技巧，比如边学边练、多问、多思考，还有两种比较通用的学习方法：打怪升级法和沉浸法。掌握了这些学习技巧，可以让学习过程事半功倍。所以，你一定要好好实践哦！

另外，你也可以把划数据结构和算法的过程，遇到什么样的困难或者疑惑吗？

欢迎留言和我分享，我会第一时间给你反馈。