

# 算法问题高效实战策略

## 01 | 如何解决问题，开启你的算法之旅？

理想的书籍是智慧的钥匙，高效的算法是成长的捷径。

你好，我是李学峰。欢迎来到“算法问题高效实战策略”学习课堂。

对很多人来说，算法和算法工程师已经不陌生。同时，对很多其他人来说，却不了解算法是什么以及算法工程师是做什么的。例如，我的家人和一些朋友就不明白我在做什么，当他们问我“什么是算法，算法工程师是做什么的”“你工作的时候是不是不能使用手机”，“需要穿西装系领带吧”，“我毕业的时候也应该像你一样去做程序员...你知道前端是怎么开发的吗”，等等，诸如此类。每当听到这些问题，可想而知，我和大家一样，心情同样是“不要不要”的。我想借此机会阐明，算法就是符合一些要求的具体步骤。

在数据科学和人工智能时代，算法工程师这一职位已然广泛存在。曾经有人告诉我，在他们家乡，小县城四五线城市数据算法工程师已经屡见不鲜。由此可见一斑。在最新的职位排行榜中，算法工程师的年薪和热度位居第一，无疑成了最受大家关注的职位之一。

如今，“算法工程师”已经包含多种类型，比如机器学习算法工程师、自然语言处理算法工程师、图像识别算法工程师、语音识别算法工程师、人工智能算法工程师、策略算法工程师、路径规划算法工程师、运筹学算法工程师，等等。看懵了是不是？我刚接触这些职业称谓时，感受和你一样！总得来说，这些不同类型的算法工程师，是为了服务不同业务而设置的。通常需要运用不同的工具，解决各自面临的问题。前五种职业名称直接挂钩“人工智能”，最受广大毕业生追捧。最后两者多使用运筹优化技术辅助进行决策，并且一般需要机器学习或统计模型的输出作为输入。策略算法作为两者的过渡，与业务的结合较为紧密。

与“程序员”不同的是，算法工程师的主要职责并非编写程序和应用系统开发。通常更聚焦于利用算法学习数据中的信息，形成决策、策略或知识。

这门专栏我不打算以上面任何一个领域的专门算法为核心，而是探讨成为一名算法工程师所必须掌握的基本算法策略。

那么，成为一个跨领域的优秀算法工程师需要具备哪些条件呢？

## 算法设计就是解决问题

工作中，算法工程师需要熟悉和了解业务，理解问题，明确目标，分析限制和可能遇到的障碍，知道痛点在哪里，会不会掉进坑里等。明白要求，限制，痛点和坑，再编写简洁高效可实现的算法才能培养一名算法工程师解决问题的能力。

练习算法与程序设计是培养解决问题的思维和能力的最佳途径。现在有各式各样的算法大赛和在线的刷题网站，例如领扣 (<https://www.leetcode.com>) 常常成为毕业生找工作面试的利器。本专栏的算法设计不同于实际研发中的要求，也将以“读入一个或一组值，然后计算出某个或某组值”的形式给出。这也更符合一般程序设计的模式，同时在解决这些问题时所做出的优化和编写的代码对实际业务也有非常大的帮助。

## 解决问题的步骤

算法设计需要丰富的知识。有人说做算法需要学习和掌握的知识太多了，一个月不学习就感觉已经 out 了。事实也确实如此！在他们脑海里，需要对数学原理、编程语言的特性、对所调用的包的功能和注意事项等进行综合思考，还需要不断给自己充电，跟踪、学习和应用最前沿的研究成果。但即便如此，我敢拍胸脯说，没有哪一个算法工程师是在掌握了所有内容后才开展工作的。

他们是怎么做的呢？

事实上，他们遵循的，正是我在开篇词中提到的方法，也是本专栏学习的中心观点：“从实战策略的角度去学习算法”。那么具体来说，到底要怎么学，学习的过程中又要特别注意哪些地方呢？

上面提到，练习算法与程序设计是培养解决问题的思维和能力的理想方式。但对于领扣等在线网站上的题目，刚开始接触往往不知如何下手，同时也容易陷入死记硬背的漩涡。解决问题的能力是一种抽象的概念，看不见也摸不着，只是一味拼命练习很难培养。就像我们小时候打篮球，大多数人在玩了一段时间后都能凑合练上几局。但最终却无法在专业的道路上更进一步，停留在了机械式运球、上篮或投篮的盲目尝试和体会中。

要成为解决问题的高手，需要更高层次的磨练。这种磨练的目的就是练就解题的技术而不是解题本身。要分析自己解决问题的方式，了解其不足，领会会在哪些地方需要改善。这个过程与练习投篮类似。所有人都知道要把篮球投进篮板，但并不是随便多次后就能提高投篮准确率。为了提高投篮实力，应该把投篮分为几个主要动作，不断检查自己是否已经掌握每个动作，对做的不正确的地方要找出改进方向。

根据我多年的学习工作经验，我把解决算法问题的重点，总结成了下面这五步，无论你是否有基础，都可以对照来做，稳步进阶。

## 第一步：问题之始，实践为根

实际问题总是纷繁复杂的，有的是痛点，有的是坑，很难找到完全一模一样的问题。我们要做的是理解问题，明确范围、数据和产出形式，并转化为算法问题。同时要明白，算法作为一门学科，却包含了丰富的内容，有应用技术，也有深刻的理论研究。如果想先掌握了一套高屋建瓴的精深理论之后再上手操作，恐怕在真要解决问题的时候，所学的大部分内容也已经忘了吧。

所以，我建议你，在掌握了必要的基础之后，就应该动手实践了。那么，在面对实际问题时，我们应该怎么做呢？很常见的做法是拿到一个任务、课题或问题之后，立马开始按照自己的理解动手码代码。但在最后结题和产出时却发现与需求背道而驰，最终结果也可想而知了。

正确的做法是拿到一个问题，第一步应该理清这是一个什么问题，所需数据、限制是什么？需要按照什么节奏给出什么样的产出，等等。有没有人过去遇到过，如果遇到过，他们是怎么解决的？我能复用和改进他们的解决方式吗？明确了这些问题，动起手来才会事半功倍。

## 第二步：解决方案，决胜之本

制定一个好的解决方案能让你顺利地达成目标，在提高产出的同时提升自信心。对一个算法来说，解决方案是解决问题的框架和具体可执行的步骤。其中包含选择什么样的具体方法，算法和数据结构等。这一阶段是整个算法的核心。有可能非常简单，也有可能会非常棘手。

## 第三步：验证方案，质量保证

拥有解决方案并不意味着马上就可以动手去实现，还需要验证算法的正确性。证明算法是否考虑周到，能否在所有情况下正确运行。还是只能解决问题的某些部分，在时间和效率上也不尽如人意？

当然，除了苦行僧似的冥思苦想去证明算法的正确性，请同事评议，与同事交流讨论也不失为一种好的策略。

## 第四步：实现方案，能力担当

无论多么精妙的算法，如果实现不正确或效率低下，将会导致程序无法正确运行，离产出遥遥无期。因此如果解决方案是一座大厦的骨架，那么实现它

的方式就决定了这座大厦最终能否美轮美奂。实现方式的稍微不同都会导致效率的千百倍差异！

## 第五步：温故知新，更进一层

在这里，问题已经基本上解决了。不过，通过与其他人交流讨论，请同事提建议和让他们表达自己的想法，却有可能摸索出更好的解决方案。因此，除了将自己成功和失败的经验总结成册以外，多与他人交流学习，多看看他人看待和解决同一问题的角度和思路，常常也会有意想不到的收获。

### 解决问题的策略

能否通过简单的方式解决问题。曾经有一位计算机专业的同事告诉我，他们的算法都很简单，就是直觉。同事的这番话启发了我，后来每遇到一个问题，我都优先考虑采用简单直观的解决方式。诚然，实践中的算法，不管最后呈现出来有多么精妙和复杂，一般来说，它最初的版本只简单实现了功能，经过逐步改进而迭代至今。

能否转化问题。有些问题，通常与以前解决过的问题类似，这时可以考虑复用以前的解决方案。这样就形成了对这一类问题的成型解决方案。

能否简化问题。如果一个问题太复杂，我们可以尝试将它简化为一个容易的版本，如去掉约束、减少变量、高维变一维等。

能否分解问题。分解问题同样是将复杂问题转变成简单形式的一种策略。例如把复杂的制约条件分解成若干个简单条件组成的集合。再分别解决各个问题。

能否公式化问题。我在学生时代接受的训练大都属于这个范围，即严密推导出问题的答案。如果我们能把一个问题的答案用明确的公式和步骤表达出来，何乐而不为呢？千行万行代码也不如一个公式简洁优美啊！

能否画成图。图形是理解问题最直观的方式之一，即使画图不能直接解决问题，也能帮助我们直观感受问题，从而为解决问题找到突破口。

能否只用特定形式解决问题。人类的思维特点之一是对纷繁复杂的事物进行归类。在解决算法问题时也可以把形态不同但结果相同的答案归纳到一个集合，只需解决集合中具有代表性的个体即可。

今天，我跟你分享了算法的学习方法和注意事项，其实这些观点不只适用于算法，也有助于你学习其它领域的知识，希望你能牢记在心。在接下来的课程里，我会带你逐步突破，最终成为一名算法高手。

那么，对于算法学习，你有什么困扰或是心得吗？打开微信，点击右上角“添加+”，搜索“公众号”，输入“算法情报局”，点击“关注”，回复你的问题，让我来和你一起看一看，帮你分析和解决它。

投资自己，实用的算法课，我们下期再见！