

00 开篇词 为什么大厂都爱考动态规划？

你好，我是卢誉声，很高兴能在这个专栏与你见面，和你一起搞定动态规划。

开门见山，我先做一个自我介绍。最开始，我在思科系统（Cisco Systems）工作，曾参与设计和开发了下一代视频会议系统的核心数据交换服务。我的工作涵盖了协议栈开发、微服务设计、分布式系统编配以及弹性算法设计。

这段经历让我形成了一个认知：算法对设计关键服务来说十分重要，它决定了系统的稳定性、弹性以及可扩展性。

后来，我加入了 Autodesk，成为了一款三维设计旗舰软件的框架和平台软件工程师。负责开发了基于大规模结构化数据的高性能搜索引擎，首次将灵活的多线程和异步框架带入产品框架层面，在原有的底层内存模型上采用了改进后的检索引擎，相较于原有的搜索功能，实现了超过 300 倍的性能提升。除此之外，我还改进并维护了用于改进用户体验的数据处理系统，在平台框架层面的工作，让我积累了大量的工程实践经验。

现在，我在 Autodesk 数据平台就职，负责设计和开发大规模数据的分析、丰富化以及流化分布式服务。

我发现自己的职业发展一直围绕着数据在不断前进。基于此，我常说的一句话是：“**数据即是正义**”。

那直到今天，我的态度依然没有变。数据为媒，算法为介，而在极其重要的算法中，动态规划其实占了很大的比重。

数据为媒，算法为介
动态规划即是正义



事实上，如果你平常关注大厂面试的话，你会发现，但凡是研发岗位，无论是招聘初级还是高级工程师，大厂都倾向于安排一轮或多轮专门的算法面试环节，而且在面试环节提出动态规划相关问题的这种趋势已经愈发明显。

这是为什么呢？我来谈谈我的看法。

先说算法这件事吧。我想请你回想一下，当处理数据结构相关的问题时，你有没有这样的经历？

1. 你本能地到工具函数或者库函数中寻找有没有现成的工具。如果问题得到快速解决，它是不是迅速就成了过眼云烟？
2. 如果这个问题看起来比较棘手，它不是一个典型的算法问题，那么就寻求搜索引擎的帮助，或者干脆访问 Stack Overflow 这样的“智库”寻找前人留下的解决方案？
3. 虽然平时工作中表现优异，但当你想换工作参加大厂面试时，又发现自己难以解决面试官提出的算法问题，无从下手，面对白板“望洋兴叹”？

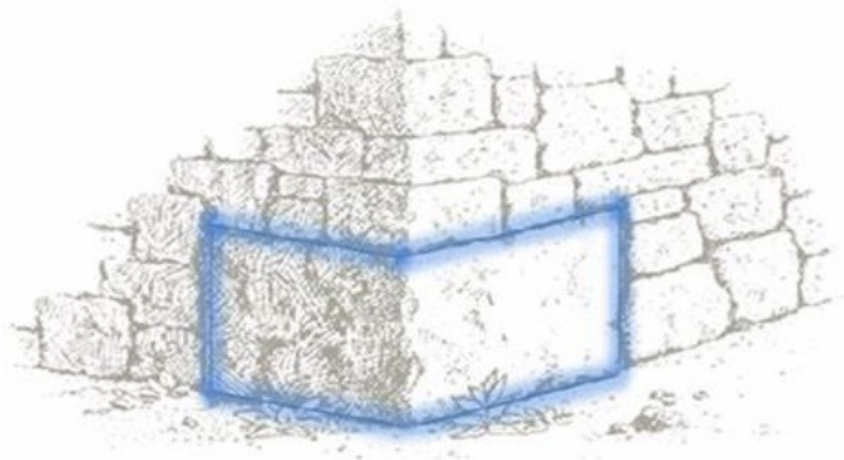
相信我，你不是一个人！这种现象很普遍。

其实，对于开发人员来说，算法和数据结构就是我们的基本功。我们常常自嘲软件研发人员的工作就是复制粘贴，搬砖就是日常工作的全部。但当公司或部门要求你去研究一个全新的技术，或者快速阅读一份开发多年且成熟的开源项目代码，并对其改造来服务于自己的产品功能时，你的压力会让你明白基本功到底有多重要！

关于基本功这事儿，我要插个故事进来，再多说几句。我曾有幸与 C++ 之父 Bjarne Stroustrup 先生进行过面对面的交流。我问了他一个问题：“如今新生代技术人员倾向于学习 Java、Go 或 Python 这些更容易上手的编程语言，您是如何看待这个现象的？” Stroustrup 先生的回答大概是这样的：“如果一个人只了解一种编程语言，那么他不能称自己是专业人士，而从我的角度上看，将 C++ 作为基础，能让你深入洞察各种各样编程语言背后的思想和设计思路。”

我觉得这个回答特别好。首先，众所周知 C++ 不是一门易学的编程语言，因此，基础不代表简单或容易。其次，C++ 能够极度自由地操纵内存资源，如果你有 C++ 的编程基础，那么在学习 Java 时就会对内存管理和控制有更深的见地。

和你分享这个小故事，当然不是强调 C++ 的重要性了，但如果你有精力学习 C++ 的话肯定能给你带来数不清的好处。其实，我这里真正想表达的就一点：**掌握好基础，能极大地拓宽我们学习更多新事物、新技术的能力。**



而算法就像技术领域的基石，它的稳定与否直接决定了大楼最终的高度。那动态规划又起到了什么作用呢？

我作为面试官曾接触过许多优秀的候选人，他们有着各种各样的背景，既有潜力又非常努力，但在面对算法问题和解决问题时没有太多思路，始终无法更上一层楼，十分遗憾。

而动态规划恰恰是解决问题的重要方法论，面对很多数据处理的应用场景，它在降低时间复杂度上极具优势，因此成为了考察重点。不仅如此，动态规划问题还能很好地考察面试者的数学

模型抽象能力和逻辑思维能力，可以反应个人在算法上的综合能力。

所以我觉得，大厂之所以如此看中一个面试者的算法基础，**特别是动态规划问题的解决能力，是因为他们更加看中一位面试者解决问题的思路与逻辑思维能力**，而不只是工具与技能的熟练程度。

讲到这儿，可能有同学会想：虽然大厂爱考，但是这东西会不会就是个绣花枕头？只是在面试中 useful，实际工作中用得上吗？

不同于普通算法，如排序或递归，动态规划从名字上看就显得很特别，有些“高端、大气、上档次”的味道在里面。但其实它离我们很近。我举个例子你就明白了，在云计算平台上一个解决方案的计算能力（容量）肯定是有限的，那么为了高效服务那些重要程度或优先级最高的客户，同时又不想浪费计算资源（说白了为了省钱），我们该怎么办？

这个问题其实可以通过队列这样的分发方式来进行一个简单的编配。但是这不够好，如果我们能够事先知道一个计算任务的重要程度和所需的计算时长，就可以通过动态规划算法来进行预演算，从数学角度推导出一个严谨的编排结果，实现有限资源的最大化利用。

你看，似乎遥不可及的动态规划问题，其实就是求最优解问题，它无时无刻都在我们身边，**总是戏剧般地提高了最优化问题的性能！**这再一次凸显出大厂为何青睐于动态规划问题，而且成为了区别面试者的一个隐形门槛。甚至可以说，掌握动态规划思想，在工作面试、技术等级晋升上都扮演了核心角色。总之一句话，动归必学。

说到这儿，估计又有同学会问：我现在知道动态规划很重要了，面试会考，工程实践要用，但问题是这玩意儿真的难啊，怎么学？

确实，如果你尝试去搜索引擎上搜动态规划的话，你会发现，检索出来的内容往往比较凌乱，很难有一个系统的方法带你从入门到精通。而像“算法面试”这样的传统书籍，对动态规划问题的描述也比较匮乏，缺乏实战经验，阅读和学习起来枯燥无味，过目就忘。

说实话，我以前也有过这种困扰。不过，近些年来在工作中用到动态规划的场景越来越多，在积累了大量的实战经验后，再结合上面面试经历，我发现还真的可以做一个较为系统的专题，针对动态规划面试题做一次深入的探讨，也就有了这门课。

我希望这个专栏，能够为你提供一个较为全面的**动态规划知识库**，兼顾理论基础和有效的经验总结，而非照本宣科的理论描述。同时，我也希望它能够为你提供一条捷径，帮助你更快地掌握动态规划问题，从容地应对面试。

为此，我精心打磨了以下三个模块。

模块一：初识动态规划

我会为你讲解复杂面试题的思考和解决方式。从贪心算法开始，一步步阐述动态规划的由来，并通过一个贯穿全篇的例子来展现动态规划的强大之处。学习和掌握这些经典的处理方法，能够为你后续掌握动态规划打下一个坚实基础。

通过这部分内容，你会系统了解到动态规划问题的特点和解题经验。

模块二：动态规划的套路

我会为你讲解动态规划问题的解题框架和套路，你可以把这个套路理解成是解决动归问题的模板。在此模板的基础上，我会向你讲解面试真题，有针对性地套用解题框架。而应对面试题的纷繁复杂，我会为你进行有效的分类，并针对每一种动态规划问题进行深入而全面的讲解。

通过这部分内容，你会快速掌握常见面试题的解题套路。

模块三：举一反三，突破套路

我会针对几种特别易考的动态规划面试题进行总结，帮助你攻破套路。并在这些高级话题的基础上，提出设计动态规划算法的关键问题。另外，还有刷题指南，所谓熟能生巧，必要的练习我们还是要的。

通过这部分内容，你会快速掌握动态规划面试题的进阶法门。

最后，我十分理解，动态规划是绝大多数人在面试中的一个老大难问题。有很多人因为它停了下来，错失了机会。掌握动态规划算法，无论是从工程实践还是面试上来说，都充满必要性。所谓大厂，带给我们的不仅仅是品牌光环那么简单，更多的是人生的一个全新阶段和崭新的台阶。

我希望，这个专栏不仅能帮你跨过大厂算法面试这道坎，还能帮你掌握一套学习复杂知识理论的思维方法，陪你度过职业发展甚至人生的重要时刻！

关于动态规划，在这里你尽可以畅所欲言，提出你的困惑和问题，非常欢迎你能与我同乘一辆车。一起加油吧！

[上一页](#)

[下一页](#)