

极客时间算法训练营

拿到 Google Offer，我的算法学习和面试经历

李煜东

《算法竞赛进阶指南》作者、Google 软件工程师

算法训练营（2021版）全新上线！

由我担任新的主讲老师

我的个人成就

竞赛经历：

2007年开始接触算法与数据结构

2010年开始大量刷题训练，1年半刷题3000+道（80%为leetcode hard++）

2012年获得全国信息学奥林匹克竞赛（NOI）金牌，国家集训队队员

大学期间获得ACM-ICPC亚洲区域赛金牌若干（平均1年2块）

2015年获得ACM-ICPC区域赛冠军，代表北京大学入选世界总决赛

我的个人成就

命题/授课经历:

2012年至今担任信息学奥赛兼职教练，在十余所学校授课，辅导学生千余人次

大学期间负责北京大学《数据结构与算法》课程期末考试机试命题

2014年担任全国信息学奥赛冬令营讲师

2015年担任全国信息学奥赛（NOI）决赛命题人、CCF学生专家

2016-2017年担任ACM-ICPC亚洲区域赛北京站命题人、裁判

2018年出版《算法竞赛进阶指南》，豆瓣评分9.2，算法竞赛选手必读书

2020年起陆续上线竞赛相关的视频课程若干

加入Google的面试经历

校招（北京）4轮算法

内容：搜索、二分、树、图、线性DP、树形DP、模式匹配、网络流

社招（加拿大）5轮算法

内容：搜索、字典树、set/map、贪心、最短路/DP，一点点多线程的内容

最近两年：

作为Google北美地区算法面试官，为面试题库供题

我是如何一步一步提升算法水平的呢？

初学算法

学算法的前三年，我大概做了200来道题，大部分是看题解、抄代码完成的

跟大家一样，很多时候处于一知半解的懵逼状态

没有抓住算法的本质，不会综合应用，甚至写不对 DFS/BFS

看到题目就想贪心 / 排序 / 找规律

有时候想出来一个思路，实现的时候也经常因为写错一个变量，少一个break，得0分

这些都是正常的 —— 你需要科学、系统的训练方法

从代码能力练起

算法面试主要考察代码能力 + 算法能力 (coding & algorithm)

- 算法能力——这题有没有思路，会不会做，解法效率如何？
- 代码能力——这题我有一个思路，能不能把程序又对又快地写出来？

尽量不抄代码，不找他人帮忙查错

尽量避免单步调试，多练习输出调试和纸上模拟调试

前期的难题coding卡住几个小时很正常，坚持自己写出来，用不了几次就能有明显提升

分类刷题

练习代码能力，模拟、搜索等题目都是不错的选择

有了一定的代码能力以后，就可以开始分类刷题，提高算法能力了

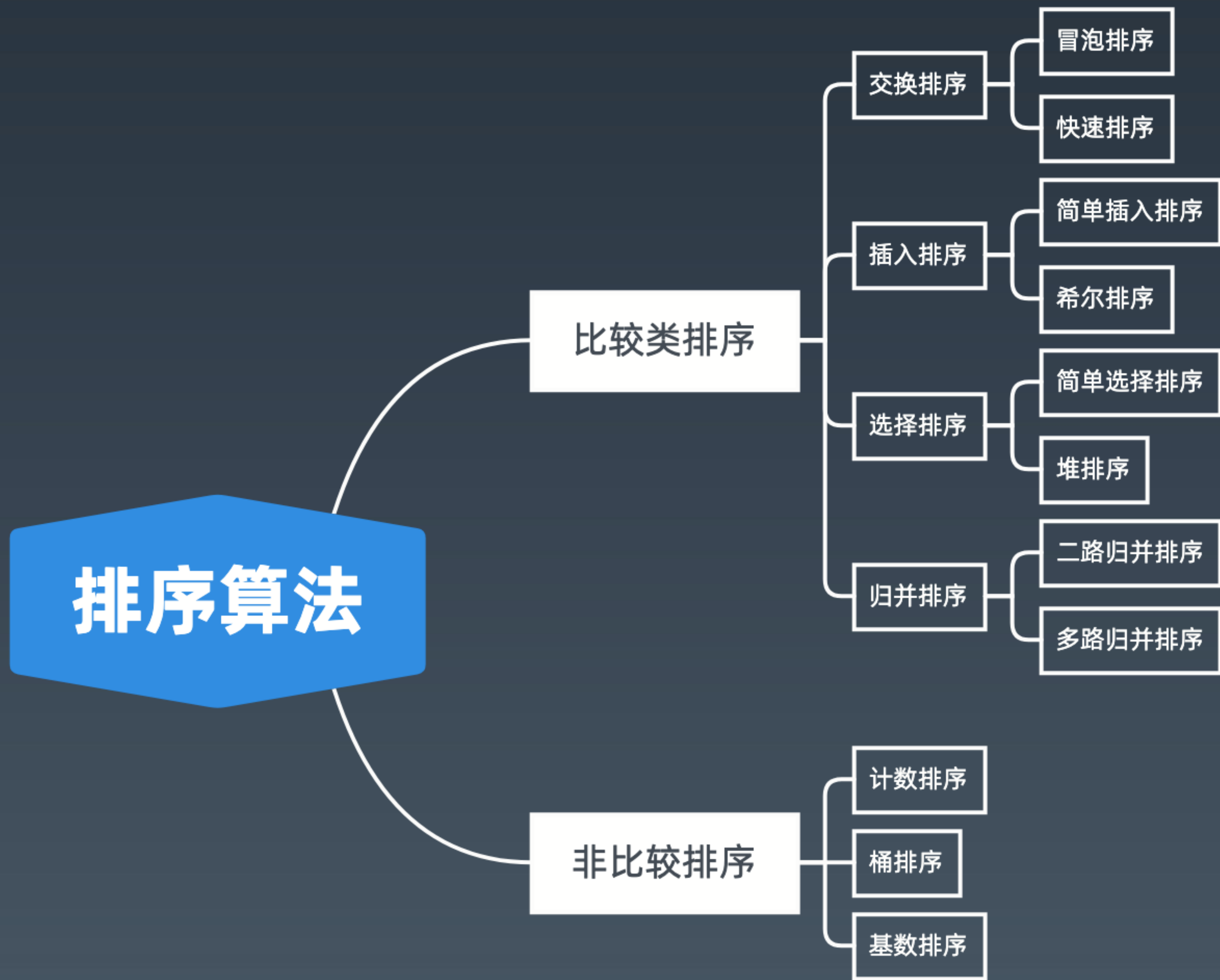
在开始系统训练后，我大概花了3个月时间提升代码能力，9个月时间分类刷题

分类刷题的主要好处

- 快速加深某一方面的知识，更方便最近学习的知识之间的类比
- 更有助于科学思维模式的形成

类比体育训练，如果说代码能力是训练体能，分类刷题就是对每一个标准动作的强化

在分类刷题的过程中，对于一个系列的知识
自己用画图、列表的方式总结对比



	稳定	不稳定	
$O(n^2)$	<p>守序划水</p> <p>插入排序</p> <p>都有地儿，一个个来嘛...</p>	<p>中立划水</p> <p>冒泡排序</p> <p>有逆序？容我交换一下...</p>	<p>混乱划水</p> <p>选择排序</p> <p>随便找个最小的？好的马上...</p>
???	<p>守序中立</p> <p>计数排序 / 基数排序</p> <p>不比较？那从数值范围入手吧</p>	<p>绝对中立</p> <p>桶排序</p> <p>我就分个组，具体咋排你们定</p>	<p>混乱中立</p> <p>希尔排序</p> <p>插排慢？这年头不会增量分组吗</p>
$O(n \log n)$	<p>守序内卷</p> <p>归并排序</p> <p>$n \log n$稳定可靠，你值得拥有 合并有序数组大家都会吧...</p>	<p>中立内卷</p> <p>堆排序</p> <p>优化也是要讲基本法的... 选最小值不考虑一下堆吗？</p>	<p>混乱内卷</p> <p>快速排序</p> <p>$n \log n$里常数最小了解一下？ swap是一门艺术，不服不要玩</p>

怎么科学刷题？

算法训练营2021之“三刷五步”训练法

科学刷题之“三刷”训练法

初学建议分类刷 -> 后期建议综合刷

一刷：

每个“小类别”的代表性题目，各刷几道
此时如果需要看题解，很正常

二刷：

复习代表性题目

“小类别”合成“大类别”，刷该分类更多的题目，举一反三，在尽量少的提示下完成

三刷：

综合性题目，尽量独立实现+测试

科学刷题之“五步”训练法

1. 理解题面
2. 部分实现
3. 有提示解答
4. 独立解答
5. 写题解

第一步 理解题面

- 理解题面的同时：
- 想一想更多的例子和测试数据，看看有没有遗漏的地方
- 提炼题目中的关键信息、变化信息

- 面试的时候，跟面试官确认自己的理解

第二步 部分实现

- 无论什么题目，先尝试实现一个朴素解法
 - 一般是搜索
 - 代码能力练好了，这一步应该不难
- 或者是部分场景下的解法
 - 尽量让自己的解法更优，覆盖更多的场景

第三步 有提示解答

- 看提示 \neq 看题解
- 这时候可以看题解的一部分，看看能否找到突破口
 - 例如题目类别，题解标题，时间复杂度，一个小结论
- 面试是一个交互性的过程，你可以与面试官交流获取适当的提示
 - 要能明白面试官在引导你什么
 - 这就要从平时练起

第四步 独立解答

- 独立完成求解，同时注意测试
- 初期训练时通常可以从第二步的搜索出发
 - 搜索时关注了哪些信息？
 - 它们有没有冗余？能不能更好地维护？
 - 有没有同类的子问题？

第五步 写题解

- 鼓励大家写题解
 - 尝试给别人讲，面试的时候也是要讲的
 - 尝试分析对比各种不同解法的优劣
 - 题解也可以写成日记的形式，记录自己遇到的难点
- 有助于加深自己的理解
- 以后也可以回看自己的题解，快速复习

关于算法训练营2021的课程

为什么要做训练营课程？

现在对于面试级别算法的教学有很多不科学之处

希望把我十余年来学习、教学算法竞赛的经验拓展到更广的人群
用更加透彻的理解、更加科学的体系帮助到你，少走弯路

训练营课程适合我吗？

20% easy, 60% medium, 20% hard

从易到难照顾各个层次的同学

不只讲知识点，重点讲一道题的思路是如何形成、如何想到的
—— 构建“思维体系”

应用落地，难写的题目，现场代码实现

2021新版算法训练营

为你整理好了每个“小分类”以及代表性题目

- 20节课，覆盖10大类、35小类知识点
- 每个知识点5~7道例题、习题，讲解不同思路

告诉你如何入手去解决一个问题

- 结合现场代码实现，教你怎样快速写出正确的代码

理论结合实战，题目覆盖面广，更新迭代及时

- 最快速度建立思维体系，与搞“题海战术”自己悟道的同学相比，达到事半功倍的效果

课程大纲一览

THANKS