

60天功课算法行动参考

第一阶段：

60天攻克算法行动的第一阶段学习目标：掌握复杂度分析，数组、栈、队列，链表，递归，排序，二分查找

📌60天攻克算法打卡行动第1天

复杂度分析是整个算法学习的精髓，只要掌握了它，数据结构和算法的内容基本上就掌握了一半。

📖学习任务是完成复杂度分析（上）：如何分析、统计算法的执行效率和资源消耗？

- 1.大O复杂度表示法
- 2.时间复杂度
- 3.空间复杂度

📌60天攻克算法打卡行动第2天

学习内容：专栏第03篇文章

📖复杂度分析（下）：浅析最好、最坏、平均、均摊时间复杂度

掌握这4个复杂度概念，我们可以更全面地表示一段代码的执行效率。

- 1.最好情况时间复杂度
- 2.最坏情况时间复杂度
- 3.平均情况时间复杂度
- 4.均摊时间复杂度

📌60天攻克算法打卡行动第3天

学习内容：专栏第05篇文章

📖数组：为什么很多编程语言中数组都从0开始编号？

数组是最基本的数据类型，我们要透过现象看本质，这4个方面会帮你理解它的精髓。

- 1.数组如何实现随机访问
- 2.低效的“插入”和“删除”
- 3.数组的访问越界问题
- 4.容器能否完全替代数组

📅60天攻克算法打卡行动第4天

📖学习内容：链表（上）：如何实现LRU缓存淘汰算法？

清理缓存通常有三种缓存淘汰策略，掌握链表的基础知识，就能实现 LRU 缓存淘汰算法。

- 1.五花八门的链表结构
- 2.链表 VS 数组性能大比拼

📅60天攻克算法打卡行动第5天

📖学习内容：链表（下）：如何轻松写出正确的链表代码？

掌握了这6个写链表代码技巧，加上你的主动和坚持，轻松拿下链表代码完全没有问题。

- 1.技巧一：理解指针或引用的含义
- 2.技巧二：警惕指针丢失和内存泄漏
- 3.技巧三：利用哨兵简化实现难度
- 4.技巧四：重点留意边界条件处理
- 5.技巧五：举例画图，辅助思考
- 6.技巧六：多写多练，没有捷径

📅60天攻克算法打卡行动第6天

📖学习内容：栈：如何实现浏览器的前进和后退功能？

后进者先出，先进者后出，掌握了“栈”的特点和应用。用两个“栈”，就能完美实现浏览器的前进、后退功能。

- 1.如何理解“栈”
- 2.如何实现一个“栈”
- 3.支持动态扩容的顺序栈
- 4.栈在软件工程中的实际应用

📅60天攻克算法打卡行动第7天

📖学习内容：队列：队列在线程池等有限资源池中的应用

队列和栈的特点正好相反：先进先出。应用它，让达到上限的资源池，实现更多请求排队。

- 1.如何理解“队列”
- 2.用数组实现的顺序队列和用链表实现的链式队列
- 3.使用循环队列避免数据搬移操作
- 4.特殊队列：阻塞队列和并发队列

🔔60天攻克算法打卡行动第8天

📖学习内容：递归：如何用三行代码找到“最终推荐人”？

递归代码难理解，写起来也很复杂。掌握下面5个问题，我们就能通过这三步：分解大问题、写出递推公式、找出终止条件，掌握递归，解决复杂问题。

- 1.如何理解“递归”
- 2.递归需要满足的三个条件
- 3.如何编写递归代码
- 4.编写递归代码时的常见问题：堆栈溢出、重复计算
- 5.怎么将递归代码改写为非递归代码

🔔60天攻克算法打卡行动第9天

📖学习内容：排序（上）：为什么插入排序比冒泡排序更受欢迎？

项目中，我们经常会用到排序。掌握下面这3种时间复杂度为 $O(n^2)$ 的经典排序算法的分析方法，我们就能高效、简单地实现小规模数据排序问题。

- 1.如何分析一个“排序算法”？
- 2.冒泡排序的原理和性能分析
- 3.插入排序的原理和性能分析
- 4.选择排序的原理和性能分析

🔔60天攻克算法打卡行动第10天

📖学习内容：排序（下）：如何用快排思想在 $O(n)$ 内查找第K大元素？

归并排序和快速排序是时间复杂度为 $O(n\log n)$ 的排序算法，掌握它们的原理和性能分析，就能实现大规模数据排序问题。

- 1.归并排序的原理和性能分析
- 2.快速排序的原理和性能分析

🔔60天攻克算法打卡行动第11天

📖学习内容：线性排序：如何根据年龄给100万用户数据排序？

桶排序、计数排序、基数排序是时间复杂度为 $O(n)$ 的线性排序算法，掌握它们的适用场景，我们就能有针对性地解决特殊数据排序问题，提高效率。

- 1.桶排序的原理和适用场景
- 2.计数排序的原理和适用场景
- 3.基数排序的原理和适用场景

🔔60天攻克算法打卡行动第12天

📖学习内容：排序优化：如何实现一个通用的、高性能的排序函数？

大部分排序函数都是采用 $O(n\log n)$ 排序算法来实现，掌握这3个方面，就能实现一个工业级的通用的、高效的排序函数

- 1.如何选择合适的排序算法
- 2.如何用优化快速排序
- 3.分析qsort() 函数的底层实现原理

🔔60天攻克算法打卡行动第13天

📖学习内容：二分查找（上）：如何用最省内存的方式实现快速查找功能？

二分查找是一种针对有序数据的高效查找算法，思想非常简单，理解了它的核心思想和适用场景，就能实现灵活应用。

- 1.无处不在的二分思想
2. $O(\log n)$ 惊人的查找速度
- 3.二分查找的递归与非递归实现
- 4.二分查找应用场景的局限性

🔔60天攻克算法打卡行动第14天

📖学习内容：二分查找（下）：如何快速定位IP对应的省份地址？

掌握这4个二分查找变形问题的解题思路，自己动手练习如何写出变体的二分查找算法，高效解决“近似”查找问题。

- 1.变体一：查找第一个值等于给定值的元素
- 2.变体二：查找最后一个值等于给定值的元素

- 3.变体三：查找第一个大于等于给定值的元素
- 4.变体四：查找最后一个小于等于给定值的元素

第二阶段：

第二阶段学习目标：散列表、二叉树、堆和堆排序、BF/RK字符串匹配算法、Trie树、图、深度/广度优先搜索

🔥60天攻克算法打卡行动第15天

🔧散列表的核心问题是散列函数设计和散列冲突解决。掌握相关基础概念和方法，就能轻松实现文本编辑器中的拼写检查功能。

- 1.如何理解散列思想？
- 2.有哪些散列函数？
- 3.散列冲突的解决方法是什么？

🔥60天攻克算法打卡行动第16天

🔧学习内容 :掌握散列表的设计方法和应用场景，我们就可以在保证散列表性能的前提下，设计出工业级水平的散列表。

- 1.如何设计散列函数？
- 2.装载因子过大了怎么办？
- 3.如何避免低效地扩容？
- 4.如何选择冲突解决方法？
- 5.工业级散列表怎么应用？

🔥60天攻克算法打卡行动第17天

🔧学习内容：散列表（下）：为什么散列表和链表经常会一起使用？

弄懂这三个例子，你就能掌握散列表和链表的结合方式，实现快速顺序遍历散列表中的数据，提高工作效率。

- 1.案例一：LRU 缓存淘汰算法
- 2.案例二：Redis 有序集合
- 3.案例三：Java LinkedHashMap

📅60天攻克算法打卡行动第18天

📖学习内容：二叉树基础（上）：什么样的二叉树适合用数组来存储？（一）

二叉树的基础知识非常重要，我们把它划分为4个部分进行学习。今天先来学习第一部分：

树和二叉树的核心概念和特征。

1.如何理解树？

2.如何理解二叉树？

📅60天攻克算法打卡行动第19天

📖学习内容：二叉树基础（上）：什么样的二叉树适合用数组来存储？（二）

今天是二叉树基础的第二部分。理解了二叉树的基础概念之后，掌握二叉树各种遍历方法的特点，我们就能选出合适的二叉树来存储数组。

1.二叉树的遍历方法一：前序遍历

2.二叉树的遍历方法二：中序遍历

3.二叉树的遍历方法三：后序遍历

📅60天攻克算法打卡行动第20天

📖学习内容：二叉树基础（下）：有了如此高效的散列表，为什么还需要二叉树？（一）

今天是二叉树基础的第三部分。二叉查找树和散列表一样，都支持动态数据集合的快速插入、删除、查找操作。通过学习这些操作，我们可以深入理解二叉查找树的特殊结构。

1.二叉查找树的查找操作

2.二叉查找树的插入操作

3.二叉查找树的删除操作

4.二叉查找树的其他操作

📅60天攻克算法打卡行动第21天

📖今天是二叉树基础的最后一部分。掌握了二叉查找树相关的操作之后，结合下面这2点，我们就能根据实际开发需求，从二叉查找树和散列表中选择最适合的数据结构，实现高效操作。

1.支持重复数据的二叉查找树

2.二叉查找树的时间复杂度分析

📅60天攻克算法打卡行动第22天

📖学习内容 :堆和堆排序：为什么说堆排序没有快速排序快？（一）

关于“堆”和“堆排序”，我们也用两天的时间进行学习。“堆”是一种特殊的树，先理解“堆”的基础概念和实现方法，再理解它的经典应用场景。

1.如何理解“堆”？

2.如何实现一个“堆”？

🏆60天攻克算法打卡行动第23天

📖学习内容 :堆和堆排序：为什么说堆排序没有快速排序快？（二）

掌握“堆”的基础概念和实现方法之后，结合“堆”排序的原理和特点，我们就能明白堆排序和快速排序之间的差异。

1.基于堆实现排序一：建堆

2.基于堆实现排序一：排序

🏆60天攻克算法打卡行动第24天

📖学习内容 :堆的应用：如何快速获取到Top 10最热门的搜索关键词？

掌握堆这3个非常重要的应用，我们就能从海量数据中快速获取热门关键词，实现搜索引擎的热门搜索排行榜功能。

1.堆的应用一：优先级队列

2.堆的应用二：利用堆求 Top K

3.堆的应用三：利用堆求中位数

🏆60天攻克算法打卡行动第25天

📖学习内容：字符串匹配基础（上）：如何借助哈希算法实现高效字符串匹配？

字符串匹配算法这部分内容，我们一共分为三部分进行学习，在第二阶段的打卡活动中，我们只学习两种基础算法：BF 算法和 RK 算法。

1.BF 算法的实现思想和应用

2.RK 算法的实现思想和应用

🏆60天攻克算法打卡行动第26天

📖学习内容：Trie树：如何实现搜索引擎的搜索关键词提示功能？

Trie 树是一种解决字符串快速匹配问题的数据结构。掌握它的实现方法和特点，我们就能利用它的优势，来实现搜索关键词提示功能。

- 1.什么是“Trie 树”？
- 2.如何实现一棵 Trie 树？
- 3.Trie 树真的很耗内存吗？
- 4.Trie 树和散列表、红黑树相比有哪些优缺点？

📅60天攻克算法打卡行动第27天

📖学习内容：图的表示：如何存储微博、微信等社交网络中的好友关系？

图是一种比树更加复杂的非线性表结构。掌握了图的基础概念和存储方法，我们就能解决复杂社交网络存储问题。

- 1.如何理解“图”？
- 2.邻接矩阵存储方法
- 3.邻接表存储方法

📅60天攻克算法打卡行动第28天

📖学习内容：深度和广度优先搜索：如何找出社交网络中的三度好友关系？

广度优先搜索和深度优先搜索是图上的两种最常用、最基本的搜索算法。了解它们的不同实现方式和适用场景，就能实现推荐“可能认识的人”这么一个功能。

- 1.什么是“搜索”算法？
- 2.什么是广度优先搜索（BFS）？
- 3.什么是深度优先搜索（DFS）？

第三阶段：

第三阶段学习目标：贪心算法、分治算法、回溯算法、动态规划、跳表、拓扑排序、Dijkstra 算法、A*算法、B+树

📅60天攻克算法打卡行动第29天

📖学习内容：贪心算法：如何用贪心算法实现Huffman压缩编码？

贪心算法是一种经典的算法思想。掌握了它的基础概念和应用，就能利用贪心算法来实现数据压缩编码，有效节省数据存储空间。

1. 如何理解贪心算法？

2. 贪心算法实战分析

🔔60天攻克算法打卡行动第30天

📖学习内容：分治算法：谈一谈大规模计算框架MapReduce中的分治思想

分治算法是MapReduce的核心思想。掌握了分治算法的基础概念和应用场景，就能指导编码来降低问题求解的时间复杂度，以及解决海量数据处理问题。

1. 如何理解分治算法？

2. 分治算法应用举例分析

3. 分治思想在海量数据处理中的应用