**算法训练营课程安排（内部版）**

所有优先掌握python版的，c++选做

[计算机应届博士生的一点求职经验——华为篇\_华为博士招聘有机考吗\_Meditator\_hkx的博客-CSDN博客](https://blog.csdn.net/Meditator_hkx/article/details/108298288)

[博士怎么找工作？ - 知乎 (zhihu.com)](https://www.zhihu.com/question/319801958)

如果一辈子都是单身，那努力是为了什么？ - 胧夜的摇篮曲的回答 - 知乎

<https://www.zhihu.com/question/402432823/answer/1364273036>

1. 高频：宽度优先搜索（BFS），深度优先搜索（DFS），二分法（Binary Search），双指针（2 Pointer），堆、栈、队列、哈希表（Heap，Stack，Queue，HashMap/HashSet），前缀和（Prefix Sum），链表（LinkedList），二叉树（Binary Tree），二叉搜索树（Binary Search Tree），快速排序与归并排序（Quick Sort/ Merge Sort）
2. 中频：动态规划（DP），扫描线（Sweep Line），字典树（Trie），并查集（Union Find），单调栈与单调队列（Monotone Stack/ Queue），TreeMap等
3. 低频：Dijkstra，树状数组，线段树，最小生成树等…

后面可以参加周赛

养成习惯先看题目描述再看视频

|  |
| --- |
| 1、算法训练营的周期为八周，采取视频+直播的方式，**每周需要大家能投入 15 小时以上的学习时间**，这样效果最佳。  2、**周一到周五需要按规划通过视频的形式学习和完成对应的习题**，每周六、周日晚上 8 点进行直播或者录播，讲解重点题和特征题，每次直播后都会提供录屏回放给大家。  3、每次直播上课讲解是 **动画 + 手写 Java 代码 或者 动画 + 手写 Python 代码**的形式，**代码不是最关键的地方，理解清楚逻辑之后你可以写出比我更加优秀的代码**。  4、为了更好的学习效果，**强烈建议大家按照我给定的刷题顺序进行刷题**。 |

|  |
| --- |
| 课程地址：[算法训练营第十二期](https://uha.xet.tech/s/3SxvFd)，课程里面的内容正在按照天数更新，暂时没有内容后者内容不全很正常。  十三期看：<https://ke.algomooc.com/p/t_pc/course_pc_detail/image_text/i_649a2d71e4b0b2d1c4292082> |

**说明**

* 绿色背景表示简单题
* 黄色背景表示中等题
* 红色背景表示困难题

1、每道题目的题解和代码既可以在飞书阅读，也可以在小鹅通(微信)上阅读。

2、每个视频的时长控制在 10 分钟左右，看完视频再去练习，养成习惯后可以 1.5 倍速看动画，代码环节正常速度看。

3、建议每天投入 **30 分钟**的时间看视频。每天大概共计投入1h

4、算法题和我们的数学题是一样的，都需要有一个先学知识点、再学例题、最后才做习题的过程，所以第一遍先把对应题目的视频看完，不要【独立】去思考、去做，否则会导致你很困扰的，效率很低。（我：所有事物的规律本应如此，应该先模仿才可能会）

5、无论自己之前是否刷过题，这一次都需要掌握视频里面教的思路、方法、代码，代码和视频动画都是完全吻合的，所以一定要先在理解动画的基础上再理解代码。

6、大家对题目遇到问题，都可以在对应的题目链接下方的评论留下你的疑惑，我来及时处理。

|  |
| --- |
| **必看视频**：[算法训练营课程使用方法✨✨✨](https://uha.h5.xeknow.com/sl/247ydz)  [ACM模式输入输出总结](https://r07na4yqwor.feishu.cn/docx/A832dKDUhox0AvxoyH7c5YNBndg) |

|  |
| --- |
| 第0周：预习前9天的概念部分 |

**第0天（周四）**

1. 7/22预习概念

|  |
| --- |
| 第一周：**数组、链表、栈、单调栈** |

**第一天（周一）7/22周六**

* 1、开营仪式
* 2、课程完整安排

开营仪式：

1. 每天1h 30min看视频只要看题跟着打代码！！ 不要自己发挥！！
2. 严格按顺序刷题，学完以后可以刷neetcode150或者neet all：<https://neetcode.io/practice> python c++都有！！
3. 用vscode leetcode插件！！刷题

装完以后问题 登陆不了<https://blog.csdn.net/qq_41883714/article/details/109639951?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-109639951-blog-125615288.235%5Ev38%5Epc_relevant_sort_base1&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-109639951-blog-125615288.235%5Ev38%5Epc_relevant_sort_base1&utm_relevant_index=2>

**第二天（周二） 7/23周日**

**数组**

数组作为最简单一个数据结构，单独考察的概率非常低，它往往是结合其它算法进行考察的。通过下面的题目帮助大家理解数组的一些基本操作。

[原地算法](https://baike.baidu.com/item/%E5%8E%9F%E5%9C%B0%E7%AE%97%E6%B3%95)in\_place algorithm

**Two-pointers:**套路：

1. 一个用来数组遍历；
2. 一个指向即将被赋值的元素！！ （**满足不同条件**!=0/!=val/!=上一个….，执行操作：移动数值/指针移向下一个！）

* [数组基础讲解](https://ke.algomooc.com/p/t_pc/course_pc_detail/video/v_64098225e4b030cacb21c49f?product_id=p_64084cf1e4b0fc5d123dbdcb&content_app_id=&type=8&parent_pro_id=p_639f0d02e4b0fc5d12185f0a) 7/20 ✔
* [LeetCode26、删除有序数组中的重复项](https://r07na4yqwor.feishu.cn/docx/OoArdoozooBkAbxYP5bcfFPBnpc)  ✔ 原地删 ∴two-pointers 套路fast指针遍历，slow指向新的
* [LeetCode 283、移动零](https://r07na4yqwor.feishu.cn/docx/KTCndS3huol5x0xsilqcFRjinng) ✔ 原地删 ∴two-pointers 同时后面置为0
* [LeetCode 485、最大连续 1 的个数](https://r07na4yqwor.feishu.cn/docx/AwMWdqs2VognNwxkn4qcRy9pnhg)  ✔ 妙啊 设置fast就是lastzero初始-1求个数
* [LeetCode 27、移除元素](https://r07na4yqwor.feishu.cn/docx/I9AldPjiBojox0xUdVlcE3zJnKf)  ✔

**第三天（周三）7/24周一**

复习昨天的过一遍10min

**链表**

Linked-list头是灵魂，head就代表了其他！

递归

1. 首先判断终止条件

C++

1. \* 定义不同类型的指针；or取内容！！
2. New
3. struct ListNode是个pointer。Pointer =value+next
4. 取next用a->next
5. Nullptr：C++11加入了nullptr，可以保证在任何情况下都代表空指针，而不会出现上述的情况，因此，建议以后还是都用nullptr替代NULL吧，而NULL就当做0使用。
6. 指针
   * 语句中 \*&ptr, &\*ptr , ptr 一样；因为 \*& 计算抵消，&\* 计算抵消, 都成 ptr 。  
     如果 声明 ptr 是指针，语句中 ptr 是指针指向的地址，\*ptr 是指针指向的 值。  
     例如：  
     int a=5;  
     int \*ptr;  
     ptr=&a;  
     printf("%d %d %d %d\n", \*&ptr,&\*ptr,\*ptr,ptr);  
     可以看出 \*&ptr,&\*ptr,ptr 输出数值一样，是a的地址。\*ptr 输出 是 a的值 5。

Python

1. None

|  |
| --- |
| 链表题目的一些知识点。自己额外定义了struct/class ListNode  1、天然的具备递归性 （就是struct体里面自己套自己）  2、设置虚拟头结点，可以起到非常好的效果 |

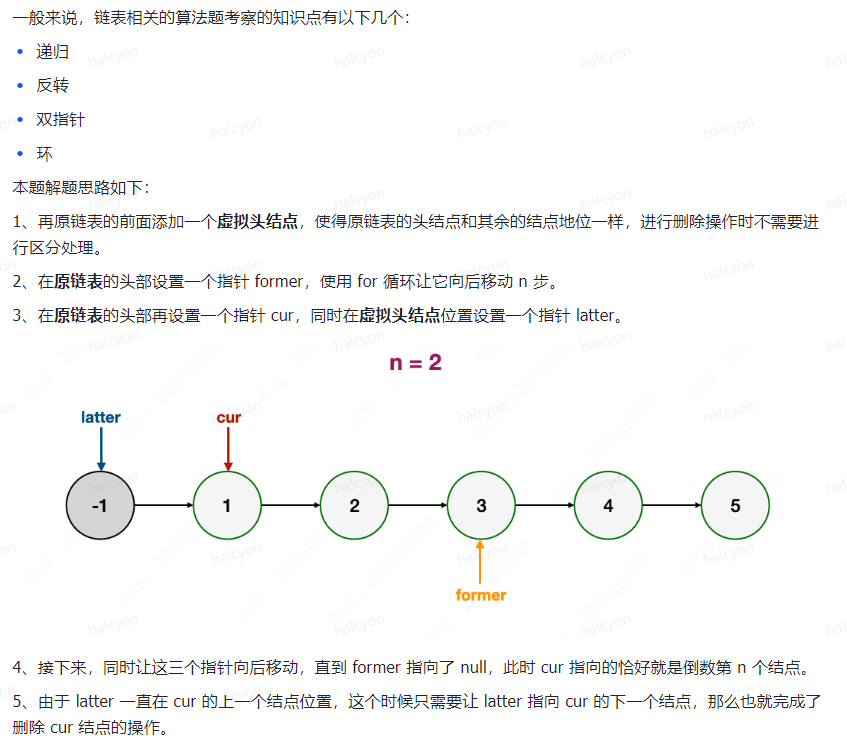
* [链表基础讲解](https://ke.algomooc.com/p/t_pc/course_pc_detail/video/v_64098223e4b02685a44fe297?product_id=p_64084cf1e4b0fc5d123dbdcb&content_app_id=&type=8&parent_pro_id=p_639f0d02e4b0fc5d12185f0a)
* [LeetCode 24、两两交换链表中的节点](https://r07na4yqwor.feishu.cn/docx/MJmbdAxP1oKC3SxUOAncGhYin7b)  ✔ 递归。对两两交换有误解？传入一个节点head当前节点，只交换它和它的下一个！
* [LeetCode 160、相交链表](https://r07na4yqwor.feishu.cn/docx/L08BdumO8ouSDSxlLDCcMTYmnwd) ✔very妙 互相走 怎么想的到的啊太妙了！！

while(\*pointA!=\*pointB){ 比内存地址，而不是比指针内容 因为\*pointA是内容 即headA结构体，结构体直接不能直接比较，有问题的<https://blog.csdn.net/Chz_1/article/details/109086388>

* [LeetCode 203、移除链表元素](https://r07na4yqwor.feishu.cn/docx/XOxAdgSQLokpV9x8osfcoWAvnKb) ✔ two-pointers; linked-list

虚拟变量 ( Dummy Variables) 又称虚设变量、名义变量或哑变量，用以反映质的属性的一个人工变量，是量化了的自变量，通常取值为0或1。引入哑变量可使线性回归模型变得更复杂，但对问题描述更简明，一个方程能达到两个方程的作用，而且接近现实。

* [LeetCode 19、删除链表的倒数第 N 个结点](https://r07na4yqwor.feishu.cn/docx/T7hldlbjZoG6ycxpaqBcqyRrnWh) ✔ 太有毒 three-pointers; former先探路如果到底了 那么和它差n步的cur就是倒数第n个；然后latter指向cur，用来删除cur而设置的



**第四天（周四）7/24周一**

路上火车刷题

**链表**

掌握：

1. 定义：value、指针
2. 初始化、对象初始化

常用技巧：

1. 当需要返回链表的时候，就常常需要dummy，虚拟表头节点dummy是新的表头节点

* [LeetCode 21、合并两个有序链表](https://r07na4yqwor.feishu.cn/docx/SoJwdKto4o0wPlxVpUFcEkMEnoc) ✔ ListNode\* dummy =new ListNode(-1) //常用的new的用法，定义新的object对象 class/struct
* [LeetCode 328、奇偶链表](https://r07na4yqwor.feishu.cn/docx/NiK8dvM6LogpvMxUcGvcchNFncc) ✔ 有点绕 就是odd even谁先动想不清楚自己
* [LeetCode 876、链表的中间结点](https://r07na4yqwor.feishu.cn/docx/Nxn6d4eGnoYeZZxFZWqcjoTVnff) ✔ 用two-pointers slow fast避免了长度判断，但是有毒，根本正常人类无法想到！！，fast步长2，slow步长1，则fast走完全长，slow走完一半
* [LeetCode 237、删除链表的节点](https://r07na4yqwor.feishu.cn/docx/JNALd5nj3o9KahxYDv7ckBqPnQb) ✔ 不懂why这个能是medium
* [LeetCode 92 、反转链表 II](https://r07na4yqwor.feishu.cn/docx/NzcjdMEIooJll9xnFmqc7TWWnhb) ❌掌握的不好4个指针，有毒；就是dummy头；pre,cur,tmp多了一个tmp； 搞清楚，断几根再连几根，注意顺序，先连谁。自己画图多试几步几个轮回。这个先断谁，没搞清楚！！！

**第五天（周五）8/3补完**

**栈**

Stack类似vector

Init:cpp: stk<type> name; python就是list

Stk.empty() Stk.top() Stk.push(a) stk.pop()

|  |
| --- |
| 1、体会**栈在【括号配对】**类型题目中的应用。  2、体会**栈在【表达式求值】**类型题目中的应用。 |

* [栈基础讲解](https://ke.algomooc.com/p/t_pc/course_pc_detail/video/v_64167c13e4b0b0bc2bc969f8?product_id=p_64084cf1e4b0fc5d123dbdcb&content_app_id=&type=8&parent_pro_id=p_639f0d02e4b0fc5d12185f0a)
* [LeetCode 20、有效的括号](https://r07na4yqwor.feishu.cn/docx/DEexdkNtPonAOwxxNCNcHygWnah) ✔
* [LeetCode 1614、括号的最大嵌套深度](https://r07na4yqwor.feishu.cn/docx/OjkSdt54QoBj2nxNPz1cOG44nMb) ✔
* [LeetCode 155、最小栈](https://r07na4yqwor.feishu.cn/docx/V9pgd90Nfoi3dsx5LJJcqFc5nSb) ✔ 栈顶是list最后面最右。就是要分类讨论very多种情况 内置stack（cpp）
* [LeetCode 150、逆波兰表达式求值](https://r07na4yqwor.feishu.cn/docx/JQmHdqzRWoxT8ExJf6KcICRunnd) ❌

c++

有点难cpp版本没有in 这个keyword；

stoi string to int

* [LeetCode 71、简化路径](https://r07na4yqwor.feishu.cn/docx/S6W8d8dRfoSkQIxaiYrckaobnTe) ✔

C++:

Stringstream: 输入输出操作[C++头文件sstream笔试常见用法\_c++ sstream\_info825的博客-CSDN博客](https://blog.csdn.net/zhendong825/article/details/124955668?spm=1001.2101.3001.6650.14&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-14-124955668-blog-108052691.235%5Ev38%5Epc_relevant_sort&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-14-124955668-blog-108052691.235%5Ev38%5Epc_relevant_sort&utm_relevant_index=15) 这是一个C++的for循环，用于遍历一个名为names的容器（例如向量、列表等）中的元素。在每次迭代中，它会将names中的当前元素赋值给名为name的变量。

const关键字表示name是一个常量引用，即在循环体内部，我们不能修改name的值。这对于遍历容器的目的非常有用，因为我们通常不希望在循环内部修改容器中的元素。

auto关键字用于自动推断变量的类型。在这种情况下，它会根据names中元素的类型来确定name的类型。

Vscode+chatgpt大法好啊，甚至都不用翻墙

typedef unsigned long long size\_t

‘’和““不一样，python就没事可以混着用

**第六天（周六）8/5补完**

直播日，直播地址：十二期<https://uha.h5.xeknow.com/sl/S7zuj>

十三期<https://ntpkq.h5.xeknow.com/sl/30SqJw>

直播课件：第一次直播.pptx

**直播笔记：**

1. 直播时复习前五天的一些基础知识点，同时讲解如下算法题。
2. 工作日是简单题；周末是中等、难题偏多！
3. **数组**：只需掌握**定义、初始化、遍历**
4. **Linked-list**：定义、经典（反转链表、环形链表、重排链表）但是往往融合了快慢指针、递归之类的知识点在链表里

**什么时候要加dummy**

* 1. 当要对原链表的头节点head操作，比如涉及删除原链表的头节点，那么头都没了，原来的整条链表都不见了有问题了，不用dummy其他办法很烦

1. 非常不建议一开始看时间、空间复杂度这种东西！！！ 因为占比很小
2. 栈

* [LeetCode 83、删除排序链表中的重复元素](https://r07na4yqwor.feishu.cn/docx/RHlLd47YaoET8ZxMvoYcJno7nqO) ✔复习
* [LeetCode 82、删除排序链表中的重复元素 II（迭代版）](https://r07na4yqwor.feishu.cn/docx/XbzidXAUJo4v7mxoy9Zc1uiNnrb) ✔why加dummy因为删除原链表头节点会有问题，头节点就不见了 重复的就删了本身也不保留
* [LeetCode 2、两数相加](https://r07na4yqwor.feishu.cn/docx/Tuhnd8295owWB0xD58NczF71nmr) ✔

very烦的一道，加法遍历，进位

现在好像很久没有真的真的让自己，静下心来，专注全力以赴做一件事情，在发表第一篇期刊以前都要逼自己前进，因为这是毕业要求，否则退学啊啊啊啊，到时候更生不如死

* [LeetCode 394、字符串解码](https://r07na4yqwor.feishu.cn/docx/DJMOdUwA4oTao1x18vVcJm5Fnye) ✔重点

C++:

Stack用.push() .top(); vector push\_back()

“”和’’不一样

* [LeetCode 224、基本计算器](https://r07na4yqwor.feishu.cn/docx/Pwc1drZTToUXGmx50Dgcv50UnMn) ✔

Python:

Char.isdigit()

Python

妙啊，可以直接调API: eval (string)= value

* [LeetCode 32、最长有效括号](https://r07na4yqwor.feishu.cn/docx/CG4idDRC3ocTwNxUW7ycMveNn4f) ✔

**第七天（周日）8/8补完7/31周一**

提前预习内容：单调栈。栈stack适合后进先出的问题

* [第五课：单调栈](https://ke.algomooc.com/p/t_pc/course_pc_detail/video/v_64167c15e4b0b0bc2bc969fd?product_id=p_64084cf1e4b0fc5d123dbdcb&content_app_id=&type=8&parent_pro_id=p_639f0d02e4b0fc5d12185f0a) 模板基本上是：for里嵌while，里面可能再嵌if

直播日，直播地址：<https://uha.h5.xeknow.com/sl/24WRHN>

十三次：<https://ntpkq.h5.xeknow.com/sl/1R7LWT>

直播课件：第二次直播.pptx

上课讲解如下算法题。

套路：

Stack；len；res结果

* [LeetCode 1475、商品折扣后的最终价格](https://r07na4yqwor.feishu.cn/docx/C1LydhT1GogybSxIundcF4P9ncg) ✔经典，while操作栈顶元素stack[-1]
* [LeetCode 739、每日温度](https://r07na4yqwor.feishu.cn/docx/KlwLdwb5LoBzSqxgeKzc9v5pneq) ✔ 和上面套路一样，也不长
* [LeetCode 84、柱状图中的最大矩形](https://r07na4yqwor.feishu.cn/docx/R4WldqtEgotjnzxW8Tmcl7LMnkd) ✔还是和第一题一样的
* [LeetCode 42、接雨水](https://r07na4yqwor.feishu.cn/docx/VzSzd7chiopSe1xkqDhclhWInRd) ✔面试高频题

|  |
| --- |
| 接雨水是究极难题，但别害怕，我可以帮你搞懂！！！如果实在不懂，暴力法都可以解决。 |

|  |
| --- |
| 第二周：队列、优先队列、哈希表、前缀和 |

**第八天（周一）8/6 7/31周一**

* [第六课：队列](https://ke.algomooc.com/p/t_pc/course_pc_detail/video/v_64167c0fe4b0b0bc2bc969f2?product_id=p_64084cf1e4b0fc5d123dbdcb&content_app_id=&type=8&parent_pro_id=p_639f0d02e4b0fc5d12185f0a)
* [第八课：优先队列](https://ke.algomooc.com/p/t_pc/course_pc_detail/video/v_64167c0ee4b0b0bc2bc969f0?product_id=p_64084cf1e4b0fc5d123dbdcb&content_app_id=&type=8&parent_pro_id=p_639f0d02e4b0fc5d12185f0a)
* [LeetCode 232、用栈实现队列](https://r07na4yqwor.feishu.cn/docx/TWvCd4EdjomLmgx0xlOcOno2nFb) ✔
* [LeetCode 641、设计循环双端队列](https://r07na4yqwor.feishu.cn/docx/HAQmdyblFoLrwfxgcKycPlJOnrg) ✔very 烦啊，太烦了这个，情况太多了！！！！这竟然是easy，我不能理解/(ㄒoㄒ)/~~
* [LeetCode 264、丑数II](https://r07na4yqwor.feishu.cn/docx/DiZwdaiLOoZMXcxyRCScVlQxnQg) ❌

乱七八糟的云里雾里

unorder\_set<data type> name; vector<data type> name; python3 heapq是库，默认最小堆 [https://blog.csdn.net/Huoon/article/details/124852302?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522169079471916800225545063%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\_id=169079471916800225545063&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~top\_positive~default-2-124852302-null-null.142^v91^insertT0,239^v12^insert\_chatgpt&utm\_term=%E6%9C%80%E5%B0%8F%E5%A0%86&spm=1018.2226.3001.4187](https://blog.csdn.net/Huoon/article/details/124852302?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522169079471916800225545063%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=169079471916800225545063&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-2-124852302-null-null.142%5ev91%5einsertT0,239%5ev12%5einsert_chatgpt&utm_term=%E6%9C%80%E5%B0%8F%E5%A0%86&spm=1018.2226.3001.4187)；

:= 海象运算符<https://zhuanlan.zhihu.com/p/107913246>

.add() 是set（）可用的

C++: greater [(103条消息) C++ std::greater用法及代码示例\_c++ greater\_诸葛灬孔暗的博客-CSDN博客](https://blog.csdn.net/qq_39383591/article/details/122360089)

* [LeetCode 373、查找和最小的 K 对数字](https://r07na4yqwor.feishu.cn/docx/BFp6dZgC7okbfZxjxRscUb0wnjd) ❌

Python：

Heappop堆操作[Python中heapq模块浅析\_heappop\_chandelierds的博客-CSDN博客](https://blog.csdn.net/chandelierds/article/details/91357784) 弹出最小值

**第九天（周二）8/6 8/1周二**

* [第九课：哈希表](https://ke.algomooc.com/p/t_pc/course_pc_detail/video/v_6416876ae4b0b0bc2bc96da0?product_id=p_64084cf1e4b0fc5d123dbdcb&content_app_id=&type=8&parent_pro_id=p_639f0d02e4b0fc5d12185f0a)

c++:

unordered\_set<data type> name; .find()查找和目标元素值相等的第一个元素;.insert(); .end()

python:

就用set()； .add(); in …

* [LeetCode 217、存在重复元素](https://r07na4yqwor.feishu.cn/docx/HUQeddvJqouPBXxI7cDccREunH8) ✔ 妙啊 不需要遍历所有的，只要有重复就停止
* [LeetCode 349、两个数组的交集](https://r07na4yqwor.feishu.cn/docx/AJn5dTKhho4xRvxZRPhc3PMHn7c) ✔这个答案不好，短的数组不应该再补全长度，不合理！！！！，反正要交集嘛！！ 代码和视频不对应，根本就没有补全！！ 三个指针;

Python:

list.sort()

cPP:

sort(vector.begin(), vector.end())

vector.back()就是最后一个元素

vector.push\_back()

|  |
| --- |
| 基于上述题目，体会**哈希集合在【判断重复元素】以及【数组去重】**上的应用。 |

* [LeetCode 1、两数之和](https://r07na4yqwor.feishu.cn/docx/SXRvdztQgoowRdxGqfscOhGunUe) ✔ auto c++：**从其初始化表达式中推导声明的变量的类型**

Unordered\_map是hash dict； unordered\_set 是hash set

* [LeetCode 219、存在重复元素II](https://r07na4yqwor.feishu.cn/docx/YOOudbkrzo4iLUxNJwIcsVKTnrp) ✔ 10min

|  |
| --- |
| 1. 基于上述题目，体会哈希表在【下标查询】类型题目中的应用 2. 另外，仔细比较 [LeetCode 217、存在重复元素](https://r07na4yqwor.feishu.cn/docx/HUQeddvJqouPBXxI7cDccREunH8) 和 [LeetCode 219、存在重复元素II](https://r07na4yqwor.feishu.cn/docx/YOOudbkrzo4iLUxNJwIcsVKTnrp) 两题，思考哈希**集合**和哈希**表**两者的相似之处和区别。 |

* [LeetCode 205、同构字符串](https://r07na4yqwor.feishu.cn/docx/LWdpdKMufotWwfxQH9PcKg3BnYe) ✔

|  |
| --- |
| 基于上述题目，体会哈希表在【元素配对】类型题目中的应用。 |

**第十天（周三） 8/2周三比较字符串：用26个字母表统计索引哈希表**

这里哈希表不用dict 了，偷懒了，用index代表字母char了用了list

补花费1h今天补周二两题， 周一两题，和上周周五一题

今天的1h

* [LeetCode 242、有效的字母异位词](https://r07na4yqwor.feishu.cn/docx/Bs7DdGr7ToS7qDxrXNucGwfenPb) ✔ 22min

Cpp

为什么记26个字母出现频率，而不是出现过的字母的频率，省事嘛？

String 用string.length();vector.size()

Vector赋值初始化：[C++:vector的初始化（5种）\_c++ vector初始化\_汐，陵的博客-CSDN博客](https://blog.csdn.net/agkl_hanhan/article/details/123427748?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-123427748-blog-87892312.235%5Ev38%5Epc_relevant_sort&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-123427748-blog-87892312.235%5Ev38%5Epc_relevant_sort&utm_relevant_index=1)

char可以直接减

Python:

str->int ord(‘a’)->int； str不可以直接减

* [LeetCode 383、赎金信（HJ81. 字符串字符匹配）](https://r07na4yqwor.feishu.cn/docx/RPtMdFIMxoar4nx6AadcjPF0nz1) ✔ 20min

和上题类似

For(auto &c: s) [for(auto &c:s)\_for(auto)\_aaaolivia的博客-CSDN博客](https://blog.csdn.net/aaaolivia/article/details/90053352)；引用&是一个[指针变量](https://so.csdn.net/so/search?q=%E6%8C%87%E9%92%88%E5%8F%98%E9%87%8F&spm=1001.2101.3001.7020)，它指向一个已经存在的变量。引用的值是指针指向的内存地址，可以通过引用来修改变量的值 所以可以通过指针c修改s的元素； 不加&那么对c的操作不影响s本身 这里也可以不加&

* [LeetCode 387、字符串中的第一个唯一字符](https://r07na4yqwor.feishu.cn/docx/EO7TdeFUaoyBhEx3wamcAbEOnIe) ✔ 20min

Cpp:

Unordered\_map<char,int>. counter;

Python:

Counter内置函数 return dict

* [LeetCode 409、最长回文串](https://r07na4yqwor.feishu.cn/docx/Qg1PdsZh6o5AhMxoSUpc7tjnn9g) ✔ 20min

长度为什么58？ 因为65-122

C++：

Int cnt[58]={0}

String::size\_type: 和int混用不好 ，string::size\_type它是一个无符号类型的值，而且能够存放下任何string对象的大小。另外，string类型的索引也是一个size\_type类型的。下标运算符 [ ] 接收的就是string::size\_type类型。<https://blog.csdn.net/yuanliang861/article/details/82915862>

按位与 &1 判断奇偶

|  |
| --- |
| 1. 基于上述题目，体会哈希表在【统计元素频率】类型题目中的应用。 2. 另外，请思考：如果题目已经告诉你元素的范围一定是小写字母或一定是大写字母（如 [LeetCode 242、有效的字母异位词](https://r07na4yqwor.feishu.cn/docx/Bs7DdGr7ToS7qDxrXNucGwfenPb) ），那么能否用一个长度为26的数组来代替哈希表【统计频率】的功能？ 3. 对于[LeetCode 387、字符串中的第一个唯一字符](https://r07na4yqwor.feishu.cn/docx/EO7TdeFUaoyBhEx3wamcAbEOnIe) ，如果只使用哈希表来完成的话，需要用到两次遍历。能否使用队列来优化该过程，使得整个算法只需要一次遍历？ |

**第十一天（周四）8/3**

补1h 上周周六4题

今天1.5h 太慢了

前置基础知识：[前缀和](https://r07na4yqwor.feishu.cn/docx/AEmCdb14Eomsr2xbriaceGvGnKg)

Why用前缀和？ 第二种定义方法就可不处理开头0特殊情况赋值了

* [LeetCode 303、区域和检索-数组不可变](https://r07na4yqwor.feishu.cn/docx/SgPddwXPgoIXzOxrPG5cb3cVnSd) ✔ 20min

因为查询多组，所以直接算不好，做了很多重复计算，不如用前缀和

* [LeetCode 1588、所有奇数长度子数组的和](https://r07na4yqwor.feishu.cn/docx/Y09HdeQCBoHlFTxpwbacjfoZnzc)✔ 20min
* [LeetCode 724、寻找数组的中心下标](https://r07na4yqwor.feishu.cn/docx/TrcmdbY8poysxsxy8boc3FUqnse) ✔ 20min

**第十二天（周五）8/4,5**

* [LeetCode 560、和为 K 的子数组](https://r07na4yqwor.feishu.cn/docx/UcGqdcYzuo5XAAx99nZcXJvKnHh) ✔ 20min 求和借助前缀和数组

挺难想的；

前缀和数组；找前缀和；找差值满足的次数加1

Python：<https://blog.csdn.net/qq_39478403/article/details/105746952> collections.defaultdict

* [剑指 Offer 66. 构建乘积数组](https://r07na4yqwor.feishu.cn/docx/Z6G5dV8pPoIGJ4xviShcQYgjnuc) ✔ 看的官方官网给的题解！！都是错的 我真的服了

构建左右积的辅助数组！！

* [LeetCode 1248、统计「优美子数组」](https://r07na4yqwor.feishu.cn/docx/FsG0d3NydoGPvHxgsHlcJLvgnHe)✔ 和560一样，太妙啊，就是**所有偶数0 奇数1 用按位与 &1**，那么前缀和=k

**第十三天（周六）8/5**

直播日，直播地址：<https://uha.h5.xeknow.com/sl/Gly2M>

https://ntpkq.h5.xeknow.com/sl/xEDjz

直播时复习前五天的一些基础知识点，同时讲解如下算法题。

* [LeetCode 697、数组的度](https://r07na4yqwor.feishu.cn/docx/HcTDdGyxYoiTcYxrnzYcHGiVnDe)
* [LeetCode 496、下一个更大元素 I](https://r07na4yqwor.feishu.cn/docx/CBNedFRQYoIvyjxV1yfcS3YynNg)
* [LeetCode 49、字母异位词分组](https://r07na4yqwor.feishu.cn/docx/KCn9dSkXyon9OCxLv0ccLBFwnEc)
* [LeetCode 36、有效的数独](https://r07na4yqwor.feishu.cn/docx/Rgl8dMWL3opPZFxfAJwcpYOinxh)
* [LeetCode 146、LRU缓存](https://r07na4yqwor.feishu.cn/docx/WoeAdWR7Porhc7xaQq7cl84kn1e)

**第十四天（周日） 8/6**

直播日，直播地址：<https://uha.h5.xeknow.com/sl/DvK8O>

https://ntpkq.h5.xeknow.com/sl/2lihLh

* [LeetCode 290、单词规律](https://r07na4yqwor.feishu.cn/docx/PeizdZcMeoYlv5xwEFyclA2sn0c) ✔
* [LeetCode 238、 除自身以外数组的乘积](https://r07na4yqwor.feishu.cn/docx/Tq96d5UcMoQ21pxzobecygnUnFf)✔
* [LeetCode 304、二维区域和检索 - 矩阵不可变](https://r07na4yqwor.feishu.cn/docx/RhnSdtlgpoFisjxrx6McwLdfnFe) ✔类似前缀和的问题，为了避免重复计算，空间换时间，专门存从左上角到某个点的矩阵的元素和！！！
* [LeetCode 1124、表现良好的最长时间段](https://r07na4yqwor.feishu.cn/docx/TgP6d8GeLoFLAgxieKCcoOqfnxc) ❌新版没有这道
* [LeetCode 23、合并K个升序链表（优先队列思路）](https://r07na4yqwor.feishu.cn/docx/BkMIdeilxo860ox3CjAc9grbneg) ✔ heapq队queue堆heap

|  |
| --- |
| 第三周：双指针、脑筋急转弯、贪心算法 |

**第十五天（周一）8/7**

* [LeetCode 88、合并两个有序数组](https://r07na4yqwor.feishu.cn/docx/JGhtdSFlcoVzNzxGOIRcgMRrnuh) ✔

Two-pointer

why从最右开始，因为有序数组右边最大且空，所以从右边开始遍历放方便好移动

* [LootCode 26、删除有序数组中的重复项](https://r07na4yqwor.feishu.cn/docx/UqmBdCsfmo3nNtxefIAcydVcnZg) ✔ 原地删 不能用set? Day2做过
* [LeetCode 80、删除有序数组中的重复项II](https://r07na4yqwor.feishu.cn/docx/XStedYQdQoCzYZxNkFvcvdDonee) ✔
* [LeetCode 319、灯泡开关](https://r07na4yqwor.feishu.cn/docx/GbAudFghcohhJyx4Zf1cwgdknQf) ✔ 找规律!!解数学题，完全是找两次规律

**第十六天（周二）8/8**

* [LeetCode 9 、回文数](https://r07na4yqwor.feishu.cn/docx/LF4tdtfY6oymC0x8H8HckxKonnb) ✔
* [LeetCode 125、验证回文串](https://r07na4yqwor.feishu.cn/docx/AZiydmqSAokMfkxDsBlcs6bjnne) ✔ isalnum() 方法检测字符串是否由字母和数字组成

Tolower（）大写->小写

* [LeetCode 680、验证回文串II](https://r07na4yqwor.feishu.cn/docx/WjtBdQbZtowcmWxhEg2ca2mjnid) ✔

[LeetCode 143、重排链表](https://r07na4yqwor.feishu.cn/docx/NL8cd8LzuozTmYxY1AkcCO0unlb) ✔ # 快慢指针找中点；可以连等slow=fast=head

* [LeetCode 292、Nim 游戏](https://r07na4yqwor.feishu.cn/docx/WF3AdXOF3o1g1HxEmrXcQzirnEf) ✔ 数学里的找规律

**第十七天（周三）8/9**

数组操作贪心算法

利用规则不访问数组每个元素

只看局部眼前利益最大化

[小白带你学---贪心算法（Greedy Algorithm) - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/53334049)

* [LeetCode 455、分发饼干](https://r07na4yqwor.feishu.cn/docx/OnqtdjtveoVNIHxWyWUcudf2nSg) ✔典型的贪心算法

Cpp： sort(vector.begin,vector.end)

Python: list.sort()

* [LeetCode 860、柠檬水找零](https://r07na4yqwor.feishu.cn/docx/ZU2md1LVmoNElGxABxVcXtXZnKi) ✔ 分类情况很多，想清楚
* [LeetCode 605、种花问题](https://r07na4yqwor.feishu.cn/docx/KN3MdqcMUoyw5Rxqfw6cmLN8nug) ✔ 贪心算法 python or从左到右，一旦true就停了
* [LeetCode 122 、买卖股票的最佳时机 II（贪心解法）](https://r07na4yqwor.feishu.cn/docx/BGczdQTLpos91hxDRvlcV1yUn4b)✔

**第十八天（周四）8/10**

List先末端、左端排序

更新末端

* [LeetCode 452、用最少数量的箭引爆气球](https://r07na4yqwor.feishu.cn/docx/MLjOd98q6oYMhSx3KuUcP1Frnbe) ✔
* [LeetCode 435、无重叠区间](https://r07na4yqwor.feishu.cn/docx/IY8odr9t1oRwZsxicyPceIV5nrd) ✔
* [LeetCode 402 、移掉 K 位数字](https://r07na4yqwor.feishu.cn/docx/JpFzdtAq2ok4ejxGFnYc9irCnwc) ✔lstrip() 方法用于截掉字符串左边的空格或指定字符

**第十九天（周五）8/11**

求特定的 和的元素

* [LeetCode 1025、除数博弈](https://r07na4yqwor.feishu.cn/docx/Lz1tdvcWconU12xk3TscBuuDnyg) ✔找规律
* [LeetCode 15、三数之和](https://r07na4yqwor.feishu.cn/docx/Aze7d6BYVo2RivxAhUkcwrSBn6g) ✔贪心算法：局部最优之和=总体最优
* [LeetCode 16、最接近三数之和](https://r07na4yqwor.feishu.cn/docx/GwMMdH9iJojIuYx5xY8cAGZ8nrh) ✔一毛一样上面的
* [LeetCode 18、四数之和](https://r07na4yqwor.feishu.cn/docx/KpPId162No85qKx11fScgZZtnlc) ✔

**第二十天（周六）8/12**

都是双指针two-pointers，left,right

第五课时直播地址：https://ntpkq.h5.xeknow.com/sl/4xDKqS

* [LeetCode167、两数之和II- 输入有序数组](https://r07na4yqwor.feishu.cn/docx/RhRJdfBh1oziblxXqhgcgR4xnGg)✔
* [LeetCode 11、盛水最多的容器](https://r07na4yqwor.feishu.cn/docx/Whb6d7X9loghSExLVvccmqYwnse) ✔重点

代码习惯：

先写大的框架，如while for if-else之类的，再往里面填条件+执行内容

* [LeetCode 61、旋转链表](https://r07na4yqwor.feishu.cn/docx/SciPdVyQBoSy8TxPyjWcKi6UnvQ) ✔ linked-list

ListNode类；求链表长度用遍历节点+1来求

* [LeetCode 142、环形链表II](https://r07na4yqwor.feishu.cn/docx/Tnh1dqsIWoBS1DxEDDncIm4ZnVA) ✔ 判断有环——快慢指针

**第二十一天（周日）8/13**

贪心算法

DFS: [深度优先搜索(DFS) 总结(算法+剪枝+优化总结)\_HeartFireY的博客-CSDN博客](https://blog.csdn.net/yanweiqi1754989931/article/details/109603384)

第六课时直播地址：<https://ntpkq.h5.xeknow.com/sl/4ADynv>

* [LeetCode 881、救生艇](https://r07na4yqwor.feishu.cn/docx/OeFJdZexQouVv5xwxZmcBBrCnRf) （删了）✔ DFS: depth-first-search
* [LeetCode 134 、加油站](https://r07na4yqwor.feishu.cn/docx/WmsOdamY4o3CG6x0Ew2c9rk1nFb) ✔
* [LeetCode 55 、跳跃游戏](https://r07na4yqwor.feishu.cn/docx/Q8XidOsQUoK5q1x72FIcIXRdnac)✔
* [LeetCode 135、分发糖果](https://r07na4yqwor.feishu.cn/docx/NtSHdOQoJoEP76xVUCZcNx3Vn6f)✔

|  |
| --- |
| 第四周：位运算、数学题、字符串、二分法、滑动窗口 |

**第二十二天（周一）8/14**

**^=异或XOR= A’B+B’A 相同数字0 不同数字1**

任何数与其本身做异或运算结果均为0

任何数与0做异或运算结果均为其本身

* [LeetCode 136、只出现一次的数字](https://r07na4yqwor.feishu.cn/docx/KStZdIil3onGaQxxOSlcR0kMnhe) ✔找规律的题
* [LeetCode 137、只出现一次的数字 II](https://r07na4yqwor.feishu.cn/docx/HGHRduagloQgloxwoQAce0COnnc) ✔位运算>>右移=/2 没太懂
* [LeetCode 260、只出现一次的数字 III](https://r07na4yqwor.feishu.cn/docx/NqixdInW6oRPnExjeYbccxzSnUc) ❌看慢一点，求看懂不求快 提示所以是最长是32位
* [LeetCode 169、多数元素](https://r07na4yqwor.feishu.cn/docx/CB9zdGbJFoca84xX6dncQl6Onqc) ✔剑指offer 算法 分治法 divide and conquer

**第二十三天（周二）8/15**

* [LeetCode 7、整数反转](https://r07na4yqwor.feishu.cn/docx/XCz7d3E1LodZbLxzmlpcLydAngh) ✔ 耐心看完所有的
* [LeetCode 67、二进制求和](https://r07na4yqwor.feishu.cn/docx/IZg3dlNPYoPd6yxzRj4cWYEDnNg)✔
* [LeetCode 48、旋转图像](https://r07na4yqwor.feishu.cn/docx/ByXHdyrWMoyCxoxRhu9cpByKnqc) ✔ 就是找规律

并不认同，为什么这么搞，什么特定元素 相关元素 放屁扯淡，个人认为普通话太差，讲的啰嗦也不算很清楚。

勉强先看，自己带脑子思考批判去看！！

* [LeetCode 50、Pow(x,n)](https://r07na4yqwor.feishu.cn/docx/Uif1dXYb3ok13OxDEENcYKWXnkd) ✔行吧 竟然是medium

n&1=0是偶数，否则奇数 按位与

* [LeetCode 8、字符串转换整数 (atoi)](https://r07na4yqwor.feishu.cn/docx/NGgVdn7JfousHHx6qxccycsZnWd) ✔    # 测试用例有问题 tmd"-91283472332"
* [LeetCode 43、字符串相乘](https://r07na4yqwor.feishu.cn/docx/TZ5Jdg56ioVWeox6TBGc63l6ngc) ✔

**第二十四天（周三）**

在排序数组中进行二分查找 太慢了tmd搞了3-4个小时，太耗时间了，必须控制时间了开始，分清主次了

类似高中要学10门课，但是只有6门要高考 ！！that’s it

想做一堆分类讨论的数学题！！！

|  |
| --- |
| **系统学习看着两个直播回放课程。8/14**  1、[二分基础概念](https://uha.h5.xeknow.com/sl/QPCgB)✔ leetcode22, 433, 704  2、[在排序数组中进行二分查找](https://uha.h5.xeknow.com/sl/uUFkX) 没看不能快进傻逼玩意 |

* [LeetCode 704、二分查找](https://r07na4yqwor.feishu.cn/docx/LP4odUMajo9PGOxzONiciAOJnaf) ✔重点！！！
* [LeetCode 189、轮转数组](https://r07na4yqwor.feishu.cn/docx/ZPhDdaYa5oG7OEx5PW2cA3wPnDb) ✔ 别光看，要自己根据题目，笔上写然后想过程，机考就是只看题目然后敲的
* [LeetCode 33、搜索旋转排序数组](https://r07na4yqwor.feishu.cn/docx/NaJxdRsz9okfHSxe0vzcOhSfnze) ✔ trie-tree=prefix-tree前缀树字典树；这视频真的好他妈啰嗦啊！！！！
* [LeetCode 81、搜索旋转排序数组II](https://r07na4yqwor.feishu.cn/docx/Pel3drtqMoBrHJxDT32ceMnCnMH) ✔ 数字可以重复 就是
* [面试题 10.03. 搜索旋转数组](https://r07na4yqwor.feishu.cn/docx/AswCdgSh3oCX3mxLOV4cAk3UnGc) ✔ 网页版刷的
* [LeetCode 34、在排序数组中查找元素的第一个和最后一个位置](https://r07na4yqwor.feishu.cn/docx/BsQ6dey7AohOGGxDOWScmKTdnhc) ✔重点
* [LeetCode 35、搜索插入位置](https://r07na4yqwor.feishu.cn/docx/LNOBd8YJdoeamyx6piJc6lafn7f) ✔
* [LeetCode 153、寻找旋转排序数组中的最小值](https://r07na4yqwor.feishu.cn/docx/HKRmdboQKoI8Hmx9PracQeUon4f) ✔
* [LeetCode 154、寻找旋转排序数组中的最小值 II](https://r07na4yqwor.feishu.cn/docx/C9YtdB99QoA9mxx3ykwcDqbAnVh) ✔类似上面就是有重复数字 这其实还好，和上面没啥区别
* [LeetCode 410、分割数组的最大值](https://r07na4yqwor.feishu.cn/docx/FSZNdGRnyo4Hamxx1onc8mudnBh) 困难题很多是题没懂✔

再去人工智能中心实习一下，面试一下试试看

|  |
| --- |
| 1. 上述题目是最基本的二分查找题目，一定要掌握。由于排序数组具有**单调性**的性质，所以是二分查找的一个非常重点的应用场景。 2. 二分查找通常存在两种模板：**左闭右闭和左闭右开。**注意比较两种模板的细微差别，以及每一次二分查找退出循环时，指针left和right的情况和位置。 3. 循环中的**判断语句**需要尤其注意，这将直接决定退出循环时left指向哪一个元素。 4. 上面的题目数量虽然多，但思考逻辑和代码基本上都是一样，一通百通，一口气拿下全部题目。 |

**第二十五天（周四）**

**在数字中进行二分**

* [LeetCode278、第一个错误版本](https://r07na4yqwor.feishu.cn/docx/NncpdeQMmoZ7x1xm77YcCTyKnad)
* [LeetCode 69、x 的平方根](https://r07na4yqwor.feishu.cn/docx/InkhdDOWzodPZnxXgiMcKLIXnUe)
* [LeetCode367、有效的完全平方数](https://r07na4yqwor.feishu.cn/docx/EHP5dMdAVo5GkKxET3lc2DTFnch)

|  |
| --- |
| 1. 因为数字天然地呈现出递增的性质，因此在数字中进行二分查找，和在排序数组中进行二分查找的思路是类似的。 2. 要注意指针left和right的初始化，和在数组中不完全一样。 3. 要注意不同题目中，**循环中判断条件使用**。 |

**第二十六天（周五）**

**在二维矩阵中进行二分**

* [LeetCode 74、搜索二维矩阵](https://r07na4yqwor.feishu.cn/docx/MMrkd2E6govDrTxzn7rcpDWYn2g)
* [LeetCode 240、搜索二维矩阵 II](https://r07na4yqwor.feishu.cn/docx/WRqadMcZXopWdlxLdC1cXn7Sncd)

|  |
| --- |
| 1. 二维矩阵的二分查找，可能会涉及到**索引线映射**的技巧，即通过公式k = i \* m + j把二维索引i和j转换为一维索引k， 然后再对k来使用标准二分查找即可。 2. 对于 LeetCode 240、搜索二维矩阵 II 这题，直接使用LeetCode 74、搜索二维矩阵的方法肯定也能够完成，但可以思考能否使用时间复杂度更低的**双指针算法**来完成。 |

**在利用问题的二段性进行二分**

* [LeetCode 162、寻找峰值](https://r07na4yqwor.feishu.cn/docx/CxLEdc3NCoMGc3xCDYycTw8fnfd)
* [LeetCode 852、山脉数组的顶峰索引](https://r07na4yqwor.feishu.cn/docx/Uv0wdXqlZoX70OxGri0cyGcUnSc)

|  |
| --- |
| 1. 上述题目体现了二分查找的适用情况是**二段性**，即所需要的答案ans能够把数轴分成两个部分。前面出现的**单调性**，本质上也是二段性中的一种。 2. 这些题目属于二分查找中最难的那档题目之一，是因为**循环中的判断语句不好写，或是需要自己实现用来判断的函数**，但本质上仍可以套用二分查找的模板。 |

**第二十七天（周六）**

直播地址：https://ntpkq.h5.xeknow.com/sl/NfB4R

|  |
| --- |
| 本次直播为回放，邀请了一位嘉宾从另外一个角度来讲解滑动窗口的内容。 |

**滑动窗口（长度可变）**

* [LeetCode 3、无重复字符的最长子串](https://r07na4yqwor.feishu.cn/docx/S8KidiULEoUUgYxAd2ocBe7dnOV)
* [LeetCode 209、长度最小的子数组](https://r07na4yqwor.feishu.cn/docx/AGpldNRFDoP4OZxUtdQcN0LCnMb)
* [LeetCode 1695、删除子数组的最大得分](https://r07na4yqwor.feishu.cn/docx/G9judPl4io6ZfsxaWzGcHHbXnwh)
* [LeetCode 904、水果成篮](https://r07na4yqwor.feishu.cn/docx/QaOXd1QkEojKj4xIrjLcih4fnif)
* [LeetCode 76、最小覆盖子串](https://r07na4yqwor.feishu.cn/docx/BiJ1d9ZUFonQNtxxQDQcLLHFnSc)

|  |
| --- |
| 1. 基于上述题目，体会【长度可变滑动窗口】类型题目的基本思路，可以直接套用模板，请多多体会。虽然题目略多，但本质上都很相似，只需要修改少数代码即可完成。 2. 【滑动窗口】应该称之为一种技巧更合适，本质上是一种特殊的【同向双指针】算法，但由于过于其使用频率过高，所以才有了滑窗这个名字。 3. 很多滑窗的题目都会用到哈希表来辅助实现算法，哈希表一般作为【统计元素频率】的作用出现，因此练习滑窗的题目也可以提高你对哈希表的熟练度。 |

**第二十八天（周日）**

直播地址：https://ntpkq.h5.xeknow.com/sl/3bKWbr

|  |
| --- |
| 本次直播为回放，邀请了一位嘉宾从另外一个角度来讲解滑动窗口的内容。 |

**滑动窗口（长度不可变、固定的）**

* [LeetCode 643、子数组最大平均数 I](https://r07na4yqwor.feishu.cn/docx/Uenxdd3yzo7GQkxHKx1clBKLnie)
* [LeetCode 1984、学生分数的最小差值](https://r07na4yqwor.feishu.cn/docx/TepddQLBwomWXOx6pO7cWZ6SnXe)
* [LeetCode 1456、定长子串中元音的最大数目](https://r07na4yqwor.feishu.cn/docx/ZYeYdQ3CvopgBox1FB2ch7aYnkd)
* [LeetCode 1052、爱生气的书店老板](https://r07na4yqwor.feishu.cn/docx/Jm63dNbfIoZkPXxjBTZcYwRvn0g)
* [LeetCode 438、找到字符串中所有字母异位词](https://r07na4yqwor.feishu.cn/docx/W0NzdgRPQonwSqxVtb4cqSpGnLh)
* [LeetCode 567、字符串的排列](https://r07na4yqwor.feishu.cn/docx/GRStdjcuTovC8DxYcj9cySpgn7g)
* [LeetCode 239、滑动窗口最大值](https://r07na4yqwor.feishu.cn/docx/KrU3dogLFohAgIxNQLLcii12nOd)

|  |
| --- |
| 1. 上述题目均为【长度不可变滑动窗口】类型的题目。其实这类题目往往比【长度可变滑动窗口】更加简单，因为只需要始终维持窗口的左右边界之差为一个定值即可，因此在代码实现上更为简单。 2. 【长度不可变滑动窗口】类型的题目可能有简单题，相信在已经掌握了【长度可变滑动窗口】类型之后，这些题目也可以迎刃而解。注意比较两者的异同。 3. 本类题目也可以直接套用模板，请多多体会。 |

|  |
| --- |
| 第五周：递归、DFS、排序算法、分治算法、回溯算法 |

**第二十九天（周一）**

* [递归基础知识讲解](https://uha.xet.tech/s/1gge3u)
* [LeetCode 82、删除排序链表中的重复元素 II（递归版）](https://r07na4yqwor.feishu.cn/docx/ZPa1dqgksoklnqx7UWAcjkVEnqf)
* [LeetCode 206、反转链表](https://r07na4yqwor.feishu.cn/docx/AuCsdu4NNoMX7LxOaQ1clwSmnpg)

**第三十天（周二）**

* [冒泡排序基础知识讲解](https://uha.xet.tech/s/1Q4RjJ)
* [LeetCode 56、合并区间](https://r07na4yqwor.feishu.cn/docx/QWyadevw6o1fv4x2xzecxv0tnvg)
* [LeetCode 148、排序链表](https://r07na4yqwor.feishu.cn/docx/MBTddOylSooOsaxcnUjcimRanXf)
* [手撕归并排序](https://uha.xet.tech/s/4rifTM)
* [LeetCode 23、合并K个升序链表（归并思路）](https://r07na4yqwor.feishu.cn/docx/EvGldO5Uno4UASx6VvncXQ7OnGg)
* [LeetCode 4、寻找两个正序数组的中位数](https://r07na4yqwor.feishu.cn/docx/ZtApdYu0eoVnmfx1M0PcQyr9nqf)

**第三十一天（周三）**

* [手撕快速排序](https://uha.xet.tech/s/1ZG9eN)
* [LeetCode 179、最大数](https://r07na4yqwor.feishu.cn/docx/J6DXdggo0ooMV3xBPGbcQMgznEh)
* [LeetCode 215、数组中的第 K 个最大元素](https://r07na4yqwor.feishu.cn/docx/NbendEC5poRHVhxb1n0cTygxnig)
* [剑指 Offer 40. 最小的 K 个数](https://r07na4yqwor.feishu.cn/docx/QKTWdq4fFocgDnxplkjcPcqZnbb)
* [剑指 Offer 45. 把数组排成最小的数](https://ntpkq.xet.tech/s/1FXBUq)

**第三十二天（周四）**

**子集问题&&排列问题**

* [LeetCode 78、子集](https://r07na4yqwor.feishu.cn/docx/I0gYdFj6kodD0LxDVvwcNz1Dndc)
* [LeetCode 90、子集II](https://r07na4yqwor.feishu.cn/docx/NE4hdFzBaoQJoAxuWIIcriY3n8g)
* [LeetCode 46、全排列](https://r07na4yqwor.feishu.cn/docx/RkVxdL0DvoWOWwxtq41cogVAnub)
* [LeetCode 47、全排列II](https://r07na4yqwor.feishu.cn/docx/Uk3Dd9KReoZ0I6xA3xectjwvnEe)

**第三十三天（周五）**

**组合问题**

* [LeetCode 39、组合总和](https://r07na4yqwor.feishu.cn/docx/IoQfdhXfho9KNhxs8kActy33nOg)
* [LeetCode 40、组合总和II](https://r07na4yqwor.feishu.cn/docx/EjqHdur6joFLwsxMHHlccKZtnWd)
* [LeetCode 77、组合](https://r07na4yqwor.feishu.cn/docx/LdoEdp0ksoct5sxaqMScG1YpnRc)
* [LeetCode 17、电话号码的组合](https://r07na4yqwor.feishu.cn/docx/ODPOdfPBTodcarxne5RcA5zVn6e)

**第三十四天（周六）**

直播地址：https://ntpkq.h5.xeknow.com/sl/2ZifRV

**棋盘问题**

* [LeetCode 51、N 皇后](https://r07na4yqwor.feishu.cn/docx/LHHudW69oofF1ZxPl7TcBne3nEd)
* [LeetCode 37、解数独](https://r07na4yqwor.feishu.cn/docx/EXnud1kg3oMaX8xAzzKcaqwynni)
* [LeetCode 37、解数独（位运算法）](https://r07na4yqwor.feishu.cn/docx/BIwQd5RkyoUBc8xDi7EcvD4Nn5c)

**第三十五天（周日）**

直播地址：https://ntpkq.h5.xeknow.com/sl/EK6Qj

|  |
| --- |
| 本次直播为第十期内容的回放 |

* [LeetCode 200、岛屿数量](https://r07na4yqwor.feishu.cn/docx/UTRFdCpDjoQaItxJDd9cLfPrnng)
* [LeetCode 695、岛屿的最大面积](https://r07na4yqwor.feishu.cn/docx/HInYdNnOPogy2nxxlZ9cG5d9nAc)
* [LeetCode 463、岛屿的周长](https://r07na4yqwor.feishu.cn/docx/UDNXdlSW3o2wRQx9A3oc6MWqnWc)
* [LeetCode 1020、飞地的数量](https://r07na4yqwor.feishu.cn/docx/Z1PzdeZ3aolxVwxuJRfckZN1nvC)
* [LeetCode 130、被围绕的区域](https://r07na4yqwor.feishu.cn/docx/KHIrdVUaboe04qxTtkrcpGQnnpb)

|  |
| --- |
| 第六周：二叉树、BFS、堆、Top K、二叉搜索树、模拟、图算法 |

**第三十六天（周一）**

* [第十课：二叉树概念与术语](https://uha.xet.tech/s/4EJv5f)
* [二叉树的前序中序后序统一迭代法](https://uha.xet.tech/s/13WMz5)
* [LeetCode 102、二叉树的层序遍历](https://r07na4yqwor.feishu.cn/docx/FqDrdVQnloX1PsxA4YmcOL59ncb)
* [LeetCode 103、二叉树的锯齿形层序遍历](https://r07na4yqwor.feishu.cn/docx/VfmcdimMSohCvTxRwRIcavbAnHh)
* [LeetCode 105、从前序与中序遍历序列构造二叉树](https://r07na4yqwor.feishu.cn/docx/GPiMdhZKwoSzwkxIUDOcbfwxnO6)

**第三十七天（周二）**

* [LeetCode 226、翻转二叉树](https://r07na4yqwor.feishu.cn/docx/YnbwdI6WFoGxsUxxJwsccJ2on2d)
* [LeetCode 110、平衡二叉树](https://r07na4yqwor.feishu.cn/docx/Lw5Udpw3yoKMzzxEInvclF4Onpd)
* [LeetCode 654 、最大二叉树](https://r07na4yqwor.feishu.cn/docx/VapvdHD7LoZAtKxh1rec0KyVnWd)
* [LeetCode 104、二叉树的最大深度](https://r07na4yqwor.feishu.cn/docx/IdvUdYJvnonbA4x9bSickhVInXf)
* [LeetCode 111 、二叉树的最小深度](https://r07na4yqwor.feishu.cn/docx/DFVDdeDKloiQMMxxvqWceZVmnmb)
* [LeetCode 297 、二叉树的序列化与反序列化](https://r07na4yqwor.feishu.cn/docx/TqcsdJkpWoGNqcxl6L5c9CeCnRh)

**第三十八天（周三）**

* [LeetCode 199、二叉树的右视图](https://r07na4yqwor.feishu.cn/docx/Wc6QdLkR4oM7doxfMrTcc4R3npe)
* [LeetCode 114、二叉树展开为链表](https://r07na4yqwor.feishu.cn/docx/MpMXduYfMo3hhAxm8w2chd5WnUc)
* [LeetCode 222 、完全二叉树的节点个数](https://r07na4yqwor.feishu.cn/docx/XTpIdWvZxoiLKkxQ3Nic2HZHnjc)
* [LeetCode 236、二叉树的最近公共祖先](https://r07na4yqwor.feishu.cn/docx/JOUOdPjNaoJnLPx99chcsksUntd)

**第三十九天（周四）**

* [LeetCode 98、验证二叉搜索树](https://r07na4yqwor.feishu.cn/docx/XfdkdzWcUoTCOyxe7mqcitT1nBh)
* [LeetCode 530、二叉搜索树的最小绝对差](https://r07na4yqwor.feishu.cn/docx/BI5bdoHXgoSS9OxgLo8cVObAnGf)
* [LeetCode 235、二叉搜索树的最近公共祖先](https://r07na4yqwor.feishu.cn/docx/PFfydt6dlo1S9LxacZ9cb9rAnTd)
* [LeetCode 450、删除二叉搜索树中的节点](https://r07na4yqwor.feishu.cn/docx/QUyYdIofCoy3qWxV3ArczmyCnje)
* [LeetCode 538、把二叉搜索树转换为累加树](https://r07na4yqwor.feishu.cn/docx/NQH8dMupNo450Exm9Sbcx0wOnyf)

**第四十天（周五）**

* [LeetCode 1603、设计停车系统](https://r07na4yqwor.feishu.cn/docx/OmhydSLdyocXjDxSgRbcVsETnIb)
* [剑指 Offer 29. 顺时针打印矩阵](https://r07na4yqwor.feishu.cn/docx/LkR9dlJTLouE8uxdRn9c3n3Cneg)
* [LeetCode 433、最小基因变化](https://r07na4yqwor.feishu.cn/docx/QDSyd7LCTokRKWxUmTpc14spnbe)

**第四十一天（周六）**

直播回放地址：https://ntpkq.h5.xeknow.com/sl/3oejhO

|  |
| --- |
| 新增二分查找直播内容 |

* [手撕堆排序](https://uha.xet.tech/s/3vF82D)
* [LeetCode 215、数组中的第 K 个最大元素](https://ntpkq.xet.tech/s/r0obm)

**第四十二天（周日）**

|  |
| --- |
| 1. 注意比较DFS和BFS的异同。 2. **相同之处**：这两种算法都属于在树形结构或者图的搜索算法，能够访问所有的节点/位置 3. **不同之处**： 4. DFS像侦察兵一样一直优先往深处搜索；BFS像军队一样铺展开来搜索 5. DFS通常需要借助用递归实现，本质上是用到了编译栈；BFS通常需要借助队列来辅助实现 6. BFS有层的概念（level），有时候也称为波纹法，通常可以用来搜寻最短路径 7. 对于二叉树而言，DFS有先序、中序、后序三种遍历方式，但对于图而言通常没有这种分类；BFS在二叉树中也称为层序遍历。 8. 图可以有多用多种方式来表示，譬如**二维矩阵、邻接矩阵、邻接表**等等，但本质上都大同小异。不要拘泥于图的表示形式，重点还是要理解DFS和BFS的过程以及万能模板。 9. DFS和BFS的应用很多，不仅仅在这种显式的图的问题中可以用到，也可以在一些生成问题用到。这是因为可以将字符串/数组的生成过程写成一个树形结构，要注意融会贯通。 |

直播回放地址：https://ntpkq.h5.xeknow.com/sl/11h60w

* [LeetCode 743、网络延迟时间](https://r07na4yqwor.feishu.cn/docx/Xrh4dCVWQoGNpUx2RANct7fonPQ)
* [LeetCode 207、课程表](https://r07na4yqwor.feishu.cn/docx/YYqidQSiHop7a8x1EwCct3DUnDb)
* [LeetCode 210、课程表II](https://r07na4yqwor.feishu.cn/docx/R0AwdY8P5on3eKximIkc1HGDnuc)
* [LeetCode 133、克隆图](https://r07na4yqwor.feishu.cn/docx/Dzz7dpAnKoMaBUxtzK7cStjHnvd)
* [LeetCode 785、判断二分图](https://r07na4yqwor.feishu.cn/docx/TlBPdTgd2okJGYxwluJcXxVbnib)
* [LeetCode 684、冗余连接](https://r07na4yqwor.feishu.cn/docx/BMBldlDnLoGZ1mx2LcPc0e0gn0f)

|  |
| --- |
| 第七周：动态规划 |

**第四十三天（周一）**

**动态规划（序列DP）**

**必看视频**

[从零开始学动态规划](https://ntpkq.h5.xeknow.com/sl/2Z6XlG)

**入门问题（理解DP基础概念）**

* [LeetCode 509、斐波那契数](https://r07na4yqwor.feishu.cn/docx/ShgEdS1lToL2RDx6Y4Kcm1MGndb)
* [LeetCode 70、爬楼梯](https://r07na4yqwor.feishu.cn/docx/GpUhd62i9oLB1WxYiVdcDoeDnih)

|  |
| --- |
| 1. 斐波那契数列是一个非常好的题目，既与递归有关，也与DP有关。可以从这个简单问题，思考**递归与DP的内在关联。**  * 递归往往是从后往前计算 * DP往往是从前往后计算  1. 用DP解题，一般要思考三个重要问题。 2. DP数组的定义是什么？ 3. 动态转移方程是什么？ 4. 如何对DP数组进行初始化？   解答了这三个问题，代码基本上呼之欲出。千万不要拘泥于复杂的定义，要**重理解而轻概念。** |

**路径问题**

* [LeetCode 62、不同路径](https://r07na4yqwor.feishu.cn/docx/YoKwdAzT0oLVD3x21j2cJ4eznZg)
* [LeetCode 63、不同路径II](https://r07na4yqwor.feishu.cn/docx/EN1pdXWttogsHUxCyn0cFn4Rnje)
* [LeetCode 64、最小路径和](https://r07na4yqwor.feishu.cn/docx/EXlcd92AroS0o3xVbrxcgRW0ndt)

|  |
| --- |
| 1. 路径问题是一类**非常典型的序列DP问题，**其过程简单、思路直接、变式不多，不同路径问题的基本思路相似，非常适合入门DP算法的学习。 2. 以LC62. 不同路径为种子题的二维路径问题，本质上大同小异，仅需在种子题的加以简单的修改，即可完成多道题目。 3. 对于路径问题而言，移动方向往往是具有限制条件的，譬如从**矩阵的左上方移动到右下方，只能向下和向右移动**等等，如果不加这种限制条件，那么较难使用DP算法解题，而应该转而考虑DFS/BFS等搜寻算法来完成。 |

**第四十四天（周二）**

下方真题视频地址：https://ntpkq.h5.xeknow.com/sl/293X1a

**最长递增子序列（LIS）问题**

* [LeetCode 300、最长递增子序列](https://r07na4yqwor.feishu.cn/docx/TS1WdWRMEoYTLDxKAXpcQbXYnrf)
* [LeetCode 673、最长递增子序列的个数](https://r07na4yqwor.feishu.cn/docx/KRhcdq7SToquRXxttGDcrF8Tneh)
* [LeetCode 334、递增的三元组](https://r07na4yqwor.feishu.cn/docx/J7tZdltjVorMAxxKuXxc5kXtnic)

|  |
| --- |
| 1. **LIS问题**是非常经典的序列DP问题，其DP解法不难理解，同时属于高频考题，强烈建议掌握。 2. LIS问题还存在更优的**贪心+二分查找**的解法，属于较难想到、较难理解的思路，感兴趣且学有余力的话可以学习一下。通常而言，DP解法已经足够解决10^4这个数据量级的题目了。 |

**最长公共子序列（LCS）问题**

* [LeetCode 718、最长重复子数组（HJ75. 公共子串计算）](https://r07na4yqwor.feishu.cn/docx/QozTd5R3EohdvAxKsdOcbZhSnvc)
* [LeetCode 1143、最长公共子序列](https://r07na4yqwor.feishu.cn/docx/PN0PdzcH0oE4K6xHbdxcoagpnyd)

|  |
| --- |
| 1. **LCS问题**是OD考试非常高频的考点，其难点在于dp数组的构建与定义较难想到。**其实用一个二维dp数组来表示两个序列之间的关系**，这是一种非常常用的技巧，但如果从来没有见过这样的做法是很难想到应该这样完成的。 2. LC1143. 最长公共子序列和LC718. 最长重复子数组两道题之间的差别仅仅在于序列是否可以连续地取，在思路和代码上是非常类似的。 3. 如果觉得理解上有些困难，那么这类题是**可以直接背诵代码**的，因为代码不长。 |

**第四十五天（周三）**

**动态规划（状态DP）**

|  |
| --- |
| 一定要按照下面的顺序做题！！！完完整整的看完直播回放！！！ |

* 直播回放：https://ntpkq.h5.xeknow.com/sl/1J5ef3
* [LC188. 买卖股票的最佳时机 IV](https://r07na4yqwor.feishu.cn/docx/OPQDd82lLop2VfxEuShceh9vnCg)
* [LeetCode 121、买卖股票的最佳时机](https://r07na4yqwor.feishu.cn/docx/H6ACdXECGoS5aBxRrcacVV8Tnob)
* [LeetCode 123、买卖股票的最佳时机 III](https://r07na4yqwor.feishu.cn/docx/YfKcds03BozhyTxauQ9ceRxInod)
* [LeetCode 122、买卖股票的最佳时机 II](https://r07na4yqwor.feishu.cn/docx/DhvHd1J6Wo50grxOUoRcnATwntJ)
* [LeetCode 309、最佳买卖股票时机含冷冻期](https://r07na4yqwor.feishu.cn/docx/RGCCdHAL7oMp0NxLto6clPdOnYb)
* [LeetCode 714、买卖股票的最佳时机含手续费](https://r07na4yqwor.feishu.cn/docx/NYMZdH2C3oKK6ZxDdUVc0KMbn0d)

|  |
| --- |
| 1. 股票问题是非常经典的一类**状态DP**问题。所谓状态DP，就是dp数组通常定义为一个二维数组，而第i个位置，通常包括多种不同的状态，而并非像序列DP那样只是个单一的值。 2. 状态的定义比较麻烦，稍微有点反直觉，一般来说**会将买入的状态设置为负数，表示已经花费的金钱。** 3. 该系列我会讲解一个模板解法 |

**第四十六天（周四）**

* [LeetCode 120、三角形最小路径和](https://r07na4yqwor.feishu.cn/docx/VZKzdqE8RopjFBxnpbmctOOgnxf)
* [LeetCode 343、整数拆分](https://r07na4yqwor.feishu.cn/docx/G5csdigDXoiYudx9v3xcmYJYnze)
* [LeetCode 279、完全平方数](https://r07na4yqwor.feishu.cn/docx/GKKHdXX3iojdQAxz3qQc4ZRknEg)
* [LeetCode 174、地下城游戏](https://r07na4yqwor.feishu.cn/docx/NW0SdzuVJoJlV6xpKMRcuzMrnWQ)

**第四十七天（周五）**

* [LeetCode 5、最长回文子串](https://r07na4yqwor.feishu.cn/docx/WD0mdRCFloJ2Q7xM0GJcSwK9nee)
* [LeetCode 53、最大子数组和](https://r07na4yqwor.feishu.cn/docx/IWwGd2p6xokXHcxqbUnc9c5Jnog)
* [LeetCode 516、最长回文子序列](https://r07na4yqwor.feishu.cn/docx/NKmZd5z52oPtK2xitsjcfZ5EnMg)
* [LeetCode 718、最长重复子数组](https://r07na4yqwor.feishu.cn/docx/NVpDdn0z6oQ11px6YMKcG9t7nLB)

**第四十八天（周六）**

直播地址:https://ntpkq.h5.xeknow.com/sl/1oK3a8

* [LeetCode 322、零钱兑换](https://r07na4yqwor.feishu.cn/docx/PqYVdmlsNoRkN7xRRYkc7Yl5nGc)
* [LeetCode 139、单词拆分](https://r07na4yqwor.feishu.cn/docx/SXWhdD5mZow12bxHJK6co4UInsb)
* [LeetCode 264、丑数II（动态规划）](https://r07na4yqwor.feishu.cn/docx/WFAVdofhHoGt6bxSGTNcYv0qnW2)
* [LeetCode 72、编辑距离](https://r07na4yqwor.feishu.cn/docx/Px5XdN2lsoQ4bRxWkEncZmFQnUb)

**打家劫舍问题**

* [LeetCode 198、打家劫舍](https://r07na4yqwor.feishu.cn/docx/TSw5d0Woro4b14x4MQhczI4InTg)
* [LeetCode 213、打家劫舍II](https://r07na4yqwor.feishu.cn/docx/LXrzda1omoSSUixc1lzcCVwfnWh)
* [LeetCode 337、打家劫舍III](https://r07na4yqwor.feishu.cn/docx/IiindP5Aho8d4wxAerPc3h5Knvf)

**第四十九天（周日）**

直播地址：https://ntpkq.h5.xeknow.com/sl/2mS2GA

**动态规划（背包DP）**

**0-1背包**

* [0-1背包动画演示](https://ntpkq.xet.tech/s/4i6lnv)
* [0-1背包解题模板](https://ntpkq.xet.tech/s/38J6Dg)
* [0-1背包优化之滚动数组](https://ntpkq.xet.tech/s/28j9sj)
* [LeetCode 474、一和零](https://r07na4yqwor.feishu.cn/docx/Oqqhdj3nwoipwyxfSl9cUvZqnfg)
* [LeetCode 494、目标和](https://r07na4yqwor.feishu.cn/docx/YrBXdogkcoVBUMxW0bLcfoQynyG)
* [LeetCode 416、分割等和子集](https://r07na4yqwor.feishu.cn/docx/XNzTdmAbCoq7d6x9efqc84Aonib)
* [LeetCode 1049、最后一块石头的重量II](https://r07na4yqwor.feishu.cn/docx/Ezcbdm1fAo9Pa6xo5moc5XgIn8g)

**完全背包**

* [完全背包动画演示与解题模板](https://ntpkq.xet.tech/s/3uoaSj)
* [LeetCode 322、零钱兑换（完全背包解法）](https://r07na4yqwor.feishu.cn/docx/J1YLdZ6reoiqfoxZObAcWEBBnSd)
* [LeetCode 518、零钱兑换II](https://r07na4yqwor.feishu.cn/docx/TTxxd3EuooYjpcxWdW9ch5Dxnuh)

|  |
| --- |
| 第八周：并查集、Trie、补充题 |

**第五十一天（周二）**

* [什么是并查集](https://ntpkq.xet.tech/s/1cnv2F)
* [LeetCode 547、省份数量](https://r07na4yqwor.feishu.cn/docx/OmgMdcqeYoGFMxxujgOcug3qnQc)
* [LeetCode 200、岛屿数量（并查集解法）](https://r07na4yqwor.feishu.cn/docx/VJ3JdeUvHoTvMyxZwPkcVkYpnr0)

**第五十天（周一）**

* [什么是字典树](https://ntpkq.xet.tech/s/1vlfgS)
* [LeetCode 208、实现Trie（前缀树）](https://r07na4yqwor.feishu.cn/docx/JkQFdIsgNoCuvKxRcukcLAKvnCe)
* [LeetCode 211、添加与搜索单词 - 数据结构设计](https://r07na4yqwor.feishu.cn/docx/MtMVd4EG5oyelyx75p2ceBminPh)
* [LeetCode 648、单词替换](https://r07na4yqwor.feishu.cn/docx/OjRcd0OU9oTwGdxcXD2ckSv0nib)
* [LeetCode 676、实现一个魔法字典](https://r07na4yqwor.feishu.cn/docx/N6jsdjliVo05g7xmLdlcNUS3nyc)

**第五十二天（周三）**

* [LeetCode 445、两数相加II](https://r07na4yqwor.feishu.cn/docx/P6lPdYHrwodgsRxCFu0cUe65nX4)
* [LeetCode 874、模拟行走机器人](https://r07na4yqwor.feishu.cn/docx/Bj33du5FVoLcZaxuj5sc4qUEnwd)
* [LeetCode 138 、复杂链表的复制](https://r07na4yqwor.feishu.cn/docx/WlIZdYMOmo7IhZx9vJicFvWEnAb)
* [LeetCode 811、子域名访问次数](https://r07na4yqwor.feishu.cn/docx/UFpvd3p6voXXCTxFpwJcIovZn0g)

**第五十三天（周四）**

* [LeetCode 1109、航班预订统计](https://r07na4yqwor.feishu.cn/docx/J545dLYHboV3gAxm5oocq1iLnac)
* [LeetCode 45、跳跃游戏II](https://r07na4yqwor.feishu.cn/docx/TDbBdVbw6ovpeMxTcXDcpdSqnCd)
* [LeetCode 376 、摆动序列](https://r07na4yqwor.feishu.cn/docx/J6ZcdOGKnoYTMOx9z7XcDxKFnXg)
* [LeetCode 946、验证栈序列](https://r07na4yqwor.feishu.cn/docx/P7THd8inWo4vYgxvKaecuX5pnt9)

**第五十四天（周五）**

* [LeetCode 231、2 的幂](https://r07na4yqwor.feishu.cn/docx/Vppsd6fehoGP9oxr2Dwc6rznnYD)
* [LeetCode 268、丢失的数字](https://r07na4yqwor.feishu.cn/docx/M2uGdlJKbo2FPbxvOGJcxGavntS)
* [LeetCode 318、最大单词长度乘积](https://r07na4yqwor.feishu.cn/docx/YNZqdVcbLobgpBxbLvqcBAKknCf)
* [LeetCode 85、最大矩形](https://r07na4yqwor.feishu.cn/docx/CgY2dGdx8oaAt1xjePjcrfL9ns9)

**第五十五天（周六）**

直播地址：https://ntpkq.h5.xeknow.com/sl/28rQWN

* [LeetCode 461、汉明距离](https://r07na4yqwor.feishu.cn/docx/HevPdT9X5ouMi2xPqHscvVy4nug)
* [LeetCode 295、数据流中的中位数](https://r07na4yqwor.feishu.cn/docx/KMxpddjwkojCE3xZAxOcgIBYnxe)
* [剑指 Offer 51、数组中的逆序对](https://r07na4yqwor.feishu.cn/docx/O8VfdNRUPof0eHxEnW4cvRkGnpf)

**第五十六天（周日）**

* [简历、笔试、电面、面试注意事项](https://ntpkq.xet.tech/s/25TAZP)

我他妈从研三就开始找工作，投简历一直实习，一直面试一直面，面到研六 我就不行我找不到工作

|  |
| --- |
| 直冲大厂！！！ |

**以下内容作为补充，学有余力再看，持续更新中**

* [选择排序](https://ntpkq.xet.tech/s/Wrgz9)
* [计数排序](https://ntpkq.xet.tech/s/1pkln7)
* [插入排序](https://ntpkq.xet.tech/s/3v7MSh)
* [部分排序](https://ntpkq.xet.tech/s/1uOKvk)

**Comments**