# 200428

**#21 merge two sorted lists**

easy

递归解法 一遍过

**# 95. Unique Binary Search Trees II**

medium

递归 看答案，需要重点review

**#96. Unique Binary Search Trees**

Medium

递归，看答案 需要review

将n分成i、与n-i-1个左右节点

# 200429

**139. Word Break**

Medium

dp 看答案 需要review 重点在于dp和s的长度及idx的差异

# 200430

**124. Binary Tree Maximum Path Sum**

Hard

递归 一次AC

**141. Linked List Cycle**

**Easy**

链表 一次AC

**142. Linked List Cycle II**

**Medium**

链表 看原理后一次AC

<https://leetcode.com/problems/linked-list-cycle-ii/discuss/258948/%2B-python>

**143. Reorder List**

**Medium**

链表 看了答案还写了一个小时，原理可以掌握实践困难。尤其是.next=None和prev节点的应用。

# 200501

**143. Reorder List**

**Medium**

**又写了一遍**

<https://leetcode.com/problems/reorder-list/>

pre, slow.next, slow = None,None,slow.next

# pre, slow, slow.next= None,slow.next,None

# 这两句话居然是不一样的 神奇!

**1422. Maximum Score After Splitting a String**

**Easy**

<https://leetcode.com/problems/maximum-score-after-splitting-a-string/>

尝试contest，自己想的与最终solution很接近了，但没有解出来

**1423. Maximum Points You Can Obtain from Cards**

**Medium**

<https://leetcode.com/problems/maximum-points-you-can-obtain-from-cards/>

自己想的dp方法测试通过，但submit超时（真可怜）。

学到了新方法sliding window

# 200502

**437. Path Sum III**

**Easy**

这道题为什么会是easy……

递归/backtrack（进阶）

need review

# #200503

**297. Serialize and Deserialize Binary Tree**

**Hard**

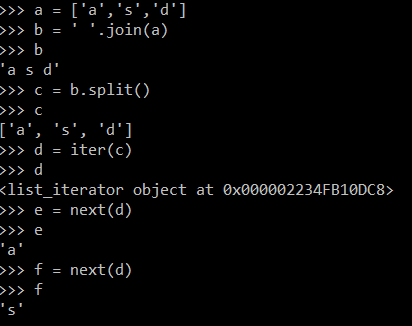
<https://leetcode.com/problems/serialize-and-deserialize-binary-tree/>

乍看不难但做了很久，三种方法搞了一下午。需要review。

可以用dfs，bfs，dict的方法

dfs麻烦的地方在于deserialize的时候如何取到合适的位置

1、如下图用iter，next每次往后移动一个idx



2、用pos=[0]的方法。实际上len(pos)一直等于1，一直使用pos[0]记录移动。与int相比递归不会返回之前的数值。

**226. Invert Binary Tree**

**Easy**

递归 自己想思路，一遍过（不愧是easy啊

# #200504

**297. Serialize and Deserialize Binary Tree**

**Hard**

<https://leetcode.com/problems/serialize-and-deserialize-binary-tree/>

重写

bfs一遍AC

dict还不是很熟，重新看了data structure后一次AC（算法思想简单）

dfs没正确的写出来，pos没弄明白。递归中pos[0]本身就记录了变化，无需在left与right之间人为改变pos。

**739. Daily Temperatures**

**Medium**

<https://leetcode.com/problems/daily-temperatures/>

自己写的方法超时了

典型的用stack比较大小的问题。stack存储的坐标，idx代表的数有从大到小排列的规律。

代码好写，算法值得思考

**278. First Bad Version**

**Easy**

<https://leetcode.com/problems/first-bad-version/>

二分查找，一次过，嘻嘻。

**1009. Complement of Base 10 Integer**

**Easy**

<https://leetcode.com/problems/complement-of-base-10-integer/>

对位运算不熟悉

# #200505

**406. Queue Reconstruction by Height**

**Medium**

看不懂题目（看懂了也做不出来），很好玩的一道题目。

代码实现的难点在于对list和dict的不熟悉。比如：

people\_sort = sorted(people, key = lambda p: p[0],reverse=True)

hs = sorted(people\_dict.keys(),reverse=True)

需要review

**387. First Unique Character in a String**

**Easy**

自己写的时间太差

**76. Minimum Window Substring**

**Hard**

双指针。噩梦一般的体验……理解原理后动手写真的好困难。

**209. Minimum Size Subarray Sum**

**Medium**

噗，经过上一题的洗礼，这题two pointers就随手写出来了。

需要注意

return length if length!=float('inf') else 0

# 200506

**647. Palindromic Substrings**

**Medium**

回文问题。忘记解法，看了答案。实现容易。

**234. Palindrome Linked List**

**Easy**

linked list问题。自己写出来，和reorder#143那道题有类似之处。但与143slow和fast确定最后slow指针的停止条件不一样（当节点个数为偶数时）。比如1，2，3，4。#143停在3处，这道题停在2处。

**9. Palindrome Number**

**Easy**

一开始想的把int转成string。后来看了去年的submission发现int可以通过余数算reversed number。

**242. Valid Anagram**

**Easy**

一次过 dict

**438. Find All Anagrams in a String**

**Medium**

sliding window，自己写AC，实现简陋。

第一遍报错是因为终止条件提前了，遗漏最尾匹配的情况。

看了discuss。用counter计数比较优雅。

# 200507

**239. Sliding Window Maximum**

**Hard**

看了花花酱的讲解<https://www.youtube.com/watch?v=2SXqBsTR6a8&t=261s>

自己写代码（很痛苦），感觉还写得挺优雅。有很多坑要注意，写在code注释里，不愧是hard题。

好的一面就是思路比较清晰，断点改错立马能反映是哪里不对。

**993. Cousins in Binary Tree**

**Easy**

自己写的，depth一开始出错。后来看discuss思路确实差不多，但实现优雅很多。

**208. Implement Trie (Prefix Tree)**

**Medium**

<https://leetcode.com/problems/implement-trie-prefix-tree/>

太恶心了这道题！搞了两个小时还是有点懵。

看了好多discuss的解法，以后还是看这次自己写的好了，比较好理解。嵌套的dict真的懵。

# 200508

**338. Counting Bits**

**Medium**

看了解答，自己写方法。

要掌握二进制数字的规律

**312. Burst Balloons**

**Hard**

dp & divide and conquer

自己没想到方法，看了花花的讲解。

实现也很麻烦，尤其是变量idx的起点和重点

**494. Target Sum**

**Medium**

自己写递归超时了

用dict方法，需要review。

# #200509

<https://leetcode.com/explore/challenge/card/may-leetcoding-challenge/535/week-2-may-8th-may-14th/3323/>

<https://leetcode.com/explore/challenge/card/may-leetcoding-challenge/535/week-2-may-8th-may-14th/3324/>

两道简单题，一次过

**128. Longest Consecutive Sequence**

**Hard**

看花花视频。思想算是hard里面比较好理解的，动手写出了蛮多问题。

**581. Shortest Unsorted Continuous Subarray**

**Easy**

两种方法，都是two pointers。感觉做题做伤了，脑子转不动了。

# #200509

**997. Find the Town Judge**

**Easy**

没意思，辣鸡题目。

**301. Remove Invalid Parentheses**

**Hard**

<https://www.youtube.com/watch?v=2k_rS_u6EBk>

hard题永远要看花花的思路。实现起来不难。但还是很伤，搞懂AC两小时。

backtracking是我非常薄弱的一块，这道题里有两个注意点，一个是去重，一个是string类型的引用如何保证回溯后恢复。这里用的恢复法是重新设置一个string，记得之前用过copy()的方法。

最近多做做backtrack。

# #200511

**22. Generate Parentheses**

**Medium**

伤心，做了好几次的题目还是懵。backtracking真是克我。

断点看了下其中递归怎么搞的，之后还是得review。

**39. Combination Sum**

**Medium**

做到暴躁，明明感觉自己写的对，但退栈也不知道那里出现的问题。对比之前的答案，发现居然是这个问题。path = path[:-1]就是不对，一直在找sum的问题，以为传值的变量递归里哪里错了。

path.pop()

优化了一下，剪枝提早排除剩余可能。

# #200512

**540. Single Element in a Sorted Array**

**Medium**

二分查找，换汤不换药。

**40. Combination Sum II**

**Medium**

backtracking 和昨天的一题差不多，但有重复。最关键的跳过重复那步没记住。

if candidates[i]==candidates[i-1**] and i>idx:**

continue

**46. Permutations**

**Medium**

没重复的backtracking，算简单的。

**47. Permutations II**

**Medium**

<https://www.youtube.com/watch?v=43w8tXWKSLw&t=1157s>

去重要记住不同方法，这题是定义一个preNum的变量记住特定位置上不重复使用某值。

46，47两题不讲求顺序，遍历时应用for num in nums:，而不用记录上一次的index。

**77. Combinations**

**Medium**

无重复组合，算是简单。

**78. Subsets**

**Medium**

无重复排列，蛮简单。backtrack的套路就是循环+递归+去重（有时）

**90. Subsets II**

**Medium**

78进阶题，加上去重。两种去重法。这一题不一定用preNum与47不同。

if i>start\_idx and nums[i]==nums[i-1]:

     continue

if nums[i]!=preNum:

     preNum = nums[i]

# 200513

**402. Remove K Digits**

**Medium**

**738. Monotone Increasing Digits**

**Medium**

两道都是贪心，无趣。

# 200514

**208. Implement Trie (Prefix Tree)**

**Medium**

上次写的很痛苦的题，掌握套路之后可以写出来了。但原理还是没有完全理解。

**37. Sudoku Solver**

**Hard**

痛苦地对backtracking有了更深的理解。

框架写出来了，一开始一直以为是backtrack part写错，后来抓脑袋debug后发现是判断函数写错了，就没有办法出现合理解。

hard还是销魂……

# 200515

**51. N-Queens**

**Hard**

backtracking。难点在于如何表示每列中queen在第几个row。用nums[i]表示，可行后，再添加至path。

**918. Maximum Sum Circular Subarray**

**Medium**

dp。需要review，分类讨论状况有点多。

**60. Permutation Sequence**

**Medium**

第一次用backtracking，dfs遍历所有可能，结果TLE，亏我还debug了好久。

<https://www.youtube.com/watch?v=xdvPD1IiyUM>

偏math题。很难搞，需要细心。

# 200516

**328. Odd Even Linked List**

**Medium**

链表，自己写出来，但操作还是不够熟，写了将近半个小时。尤其是next变动的时候，一个不注意就无限循环了。

**63. Unique Paths II**

**Medium**

dp。一次过，没什么好说的。

**72. Edit Distance**

**Hard**

dp。思路记住了，但是边缘条件没考虑好，一个是两个word可能存在空值；另一个是dp初始条件没考虑周到，最后设置了一个防止重复使用的变量才AC。

# 200517

**438. Find All Anagrams in a String**

Medium

sliding window & hash table。不难

**85. Maximal Rectangle**

Hard

三个月前做过。把二维的dp转化为一维的dp。

用stack做最小元素的挡板。注意左右范围差的计算方法。（cur\_idx-stack[-1]-1）

# 200518

**567. Permutation in String**

Medium

和昨天的438很像，换汤不换药，一次AC。没意思。

**123. Best Time to Buy and Sell Stock III**

Hard

不难。左右两次dp。了解二维dp的运行方式。

**188. Best Time to Buy and Sell Stock IV**

Hard

<https://www.youtube.com/watch?v=oDhu5uGq_ic&t=74s>

十三分钟有优化。

用这个方法写，一开始居然是memory limit，后来改成两行，TLE。

后来用了分支k>len(list)//2实现O（n），总算过了，打败21%……

做了将近三个小时，吐血。一道难忘的题目。不过做hard是有爽点的。

# 200519

**901. Online Stock Span**

Medium

没啥难度。又是和stack最后的对比弹出，循环到stack中没有比cur更小的树。

**99. Recover Binary Search Tree**

Hard

<https://www.youtube.com/watch?v=H3PPKuyzKro>

用了第一种方法，无脑inorder找异常值换value，时间不怎么好。

第二种方法，在inorder过程中直接记录node，不换list。时间好一点，但也只超过一半。

**931. Minimum Falling Path Sum**

Medium

dp。自己写的一次AC，但时间不好。

# 200520

**230. Kth Smallest Element in a BST**

Medium

这道题写了很久，一个是递归方式跳出层返回值又改变了；另一个是非递归形式还不熟练。

把最开始的constructTree方法又重写了一遍。

**94. Binary Tree Inorder Traversal**

Medium

写了94再写着道用iteration居然一次AC了，一点脑细胞都不费。（大概是上一题太伤脑筋了）

**98. Validate Binary Search Tree**

Medium

用了一种新方法，用pre记录之前的node值，一次AC。我真是小天才。

**153. Find Minimum in Rotated Sorted Array**

Medium

自己写了一遍，犯了一个错误，没搞清左边有序的条件。

# 200521

**1277. Count Square Submatrices with All Ones**

Medium

dp。学到了sum(map(sum,dp))二维数组的和

**198. House Robber**

Easy

dp。一种recursion，一种iteration，简单。

**139. Word Break**

Medium

dp。需要注意dp和s的idx差了一位。

**140. Word Break II**

Hard

dp。自己写的backtracking异常顺利，但TLE没AC。

看了答案，在递归里加了memo，过了，不算难理解。要多看看。

# 200522

**451. Sort Characters By Frequency**

Medium

题目不难。主要是要掌握counter的用法。

**347. Top K Frequent Elements**

Medium

用counter方法新做了一遍！嘻嘻，顺便复习之前写的几种方法。

**213. House Robber II**

Medium

在198基础上做了改动。

**337. House Robber III**

Medium

和之前一道max sum的题很像。看了之前的答案才做出来，需要review。

# 200523

**986. Interval List Intersections**

Medium

自己写的一次AC，厉害了。但是写了好多个ifelse，时间倒是很优越。

<https://leetcode.com/problems/interval-list-intersections/discuss/647482/Python-Two-Pointer-Approach-%2B-Thinking-Process-Diagrams>

这个比较concise

**164. Maximum Gap**

Hard

桶排序。其实不难但有好多地方卡住：

1、n-2个元素放在n-1个桶里，注定有空桶

2、gap向上取整，保证算桶idx时不会溢出

3、和最大最小值的差要记得算

4、两个数组记录每个桶的最大最小值

**174. Dungeon Game**

Hard

dp，不难，居然一次过了。

<https://www.youtube.com/watch?v=pt-xIS6huIg>