# 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**

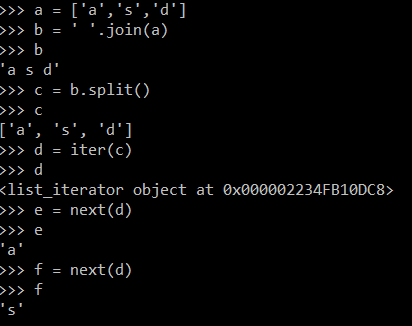
<https://leetcode.com/problems/serialize-and-deserialize-binary-tree/>

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