

1. 88: <https://leetcode.com/problems/merge-sorted-array/> (合并两个排好序的数组)

2. 921: <https://leetcode.com/problems/minimum-add-to-make-parentheses-valid/>  
○ (加入最少的括号数让当前字符括号匹配, 不过要返回新的字符串, 利口原题只用返回个数)

跟求个数一样的解法, 走一遍这个字符串记录当前开关括号的个数, 然后下面两种情况

1. 碰到开括号就开括号加一
2. 碰到关括号并且开括号大于零就将开括号减一, 开括号等于零就在结果字符串插入一个开括号

每次上面执行完了要将当前字符加入结果字符串

最后看一下开括号个数是几个, 结果字符串补上关括号

---

这道题是不是可以直接按蠡口原题先算出 "(" 每个要补几个  
然后在字符串最左边填加相应个数的 "(", 最右边添加相应个数的 ")"

---

应该可以吧, 只要能写出来就行, 复杂度是  $O(n)$

3. 207: <https://leetcode.com/problems/course-schedule/>

○ 利口 而灵妻 用 BFS & DFS 还有问 time & space complexity.

4. 311: <https://leetcode.com/problems/sparse-matrix-multiplication/> 稀疏使量的点鸡  
我面的时候电话信号不太好, 对方直接告诉我输入是压缩过的, 所以我猜应该是先  $O(n)$  来压缩向量, 类似 `[None, 1, 2, None, None, None, None, 5, 6] -> [(1, 1), (2, 2), (8, 5), (9, 6)]`。第一个是 index, 第二个是 value。

然后就是给定两个类似压缩过的输入, 用 two pointer 来 go through, 他们的 index 一样的话就乘, 不一样的话就把小的那个往后移, 循环结束后 return sum 就可以了。这步应该也是  $O(n)$ , 所以最后答案还是  $O(n)$ 。

第一题是 list

---

大概是两个向量求点积, 向量可能有很多连续相同的数, 比如 `(1, 1, 1, 2, 2, 3)`, 让你设计一个 efficient structure 去存储, 然后用这个 structure 求 dot product

--

给一个 vecto, 比如 `<1,1,3,4,7,2,5,9,10>`, 又可能 vector 很长很长。两个问题。

a. 如果 memory 有限, 怎么存储。我答可以 compress 成 string。他说 solid。

b. 给两个相同长度的 vector。形式是[(1,4), (3,1), (2,2)]这种。[(1,4), (3,1), (2,2)]代表的 vector 是<1,1,1,1,3,2,2>。求两个 vector 的 dot product。我忘了 vector 的 dot product 是啥，问了之后她解释了我就清楚了。就是对应项相乘然后加起来就是结果。我用的双指针。

--

这个就是 tag 前 50 变体呀，sparse matrix multiplication

---

题目没有说输入可能是稀疏的，对稠密向量做稀疏表示反而浪费空间。可以 RLE 压缩，点乘的时候一位一位的读出来，不用全部解压缩

5. 第二题 验证包含有撒种括弧的字符是否有笑

第二题就是验 string 是否有效

6. 两个顺序由小到大的序列 找出重复的数字

[1,2,4,5,7]

[2,5,8,9]

=> [2,5]

7. 987: <https://leetcode.com/problems/vertical-order-traversal-of-a-binary-tree/>

8. 215: <https://leetcode.com/problems/kth-largest-element-in-an-array/>

a. 第 k 大数: pque

9. 98: <https://leetcode.com/problems/validate-binary-search-tree/>

○ 验证 BST: recursion

○ 能不能优化 space complexity 的 follow up。我就说了说，也不用重新写。

10. 438: <https://leetcode.com/problems/find-all-anagrams-in-a-string/> find first string anagram's index

11. 162: <https://leetcode.com/problems/find-peak-element/> find local min, 找到一个就行，没有返回-1， similar to 利口 162

12. 680: <https://leetcode.com/problems/valid-palindrome-ii/>

○ is\_palindrome 是空格，标点符号不算进去 然后不区分大小写 今天状态很差面试一开始没读题直接写了个最简单的 is\_palindrome

13. 1249: <https://leetcode.com/problems/minimum-remove-to-make-valid-parentheses/>

○ 给一个包含"(", ")", "{", "}", "[", "]"的 string，除掉里面任意几个，使 string 合法，不要求 efficiency，不要求一定用 backtracking。

14. 211(.换成了\*): <https://leetcode.com/problems/add-and-search-word-data-structure-design/>

○ 设计 TRIE 树，两个函数: insert(word):void 和 search(word):bool

○ 不同点在于 search 的 word 可以带'\*'字符，表示任意字符

15. <https://www.1point3acres.com/bbs/thread-604200-2-1.html> 一个二维矩阵，里面的数只有 0 和 1，每一行如果有一个 1，那么这行之后的数都是 1，然后问你对于这个二

维矩阵第一个是 1 的 column. binary search 做。楼主漏了一个 edge case 就是都是 0 的情况我仍然返回了第一列，被提醒了才意识到，改过来

---

第一题最优解法为  $O(n)$

从矩阵的右上角开始判断是否为 1

If 1, then move left

Else, then move down to keep test again.

除了都是 0 的情况，只要按照题目条件，有 1 存在矩阵中，最后所在的 column 就是最左面 1 所在的 column.

线性扫描所以复杂度为  $O(n)$

---

这道题最优解是你说的这个还是 binary search 要看输入数组的形状

--

对的，从右上角往左扫描，然后往下或者往右扫描，最后复杂度  $O(n)$

---

因为如果一行中出现 1，那么这一行后面所有的数字都是 1，所以可以用 binary search 啊。我取重点，如果这一列中有 1 的话，那么我可以排除后半了，所以可以用 binary search

16. 494: <https://leetcode.com/problems/target-sum/>

○ <https://www.1point3acres.com/bbs/thread-604200-2-1.html>

○ 给一个数，比如 123，还有一个 target 每个数字前面可以加上+或者-，问多少种情况可以得到 target，比如 123 就是  $1 + 2 + 3$ ，backtrack 和 cache 可以做

○ expression add operators 的简化版

17. 238: <https://leetcode.com/problems/product-of-array-except-self/>

○ <https://www.1point3acres.com/bbs/interview/facebook-software-engineer-603816.html>

18. bst inorder traversal

19. bst iterator

20. 1026: <https://leetcode.com/problems/maximum-difference-between-node-and-ancestor/>

21. 863: <https://leetcode.com/problems/all-nodes-distance-k-in-binary-tree/>

前面讲 high level 的算法讲了很久先说了 graph 的做法面试官不喜欢 所以又临时想了树的做法 导致最后只做了这一道题

22. 3: <https://leetcode.com/problems/longest-substring-without-repeating-characters/>

○ 需要跑 test case 保证 bug free

○ 如果字符串很大很大，而且是以 incoming stream 的方式不断添加字符，如何更改你的程序让它不受内存限制

23. 273: <https://leetcode.com/problems/integer-to-english-words/> 数字 变 英语,

○ 手动了两个 tests, 用 divide and concur 没几行, 然后面试官也没让写完 hard code 的部分, 然后 test case 两个简单的+稍微长一点的简单的, 花 3min 讲清楚, 所以 code 写了将近 20min,

24. 138: <https://leetcode.com/problems/copy-list-with-random-pointer/>

Deep clone a link list

```
ListNode{
    int val;
    ListNode next;
    ListNode rand;
}
```

类似 利口 衣叁叁

关键点在于: rand 可能会链接到前面的结点, 解体思路, 把已经创建的结点存到 map 中 key:原来的结点 value:克隆的结点, 每次链接 next 和 rand 前 看看是否已经建立, 时间复杂度是  $O(N)$

第一题不是最优解

- Here march

25. 109: <https://leetcode.com/problems/convert-sorted-list-to-binary-search-tree/>

将已经排序的 link list 转化为一个 balance BTS

input: 1->2->3->4->5->6->7

output:

```
      4
     / \
    2   6
   / \ / \
  1  3 5 7
```

要求 不能用 array 解

我的解法是 recursion 找子树的根结点 (list 的中间结点)

26. 78: <https://leetcode.com/problems/subsets/>

27. 90: <https://leetcode.com/problems/subsets-ii/>

时间是  $2^N$

空间是  $O(N)$ , 因为它要求直接 print 所有的结果, 所以不需要存中间结果

28. 766: <https://leetcode.com/problems/toeplitz-matrix/>

从左下角到右上角遍历即可。可惜一开始紧张, 理解做了 Toeplitz 的定义, 只做了中间几个的检查, 几轮提示下来解法写的太 ugly, 有的时候一开始想明白了写, 比后面补救的 code 写的好很多。

---

从右下角到左上角的判断我觉得挺好的

---

呃, 这个不就是 4 行吗, 怎么会中间的几个检查这么多情况考虑?

--

是的, 我思路错了

29. 给一个数组，数组里面每一个都是一个文件名。每个文件每一行一个 string，有多行。让读所有文件，返回有重复 string 的内容和对应出现过的文件名。

○ 输入是一个数组，数组里面的元素都是文件名。每个文件中的内容按行读取。输出是让打印，所有文件中重复的内容（比较每一行看有没有完全重复的两行），和对应的文件名。如果没有重复的返回 true，有返回 false。

○ 如果确实有重复内容出现在所有的文件中，还有必要输出“对应的文件名”吗？“对应的文件名”就是所有文件呀。还是说楼主的意思是，“重复的内容”的定义是，只要有一行同时出现在两个文件中，就属于重复行，我们的目标是寻找所有的重复行，并且输出它们的源文件？

○ 是的，只要出现在两个文件中的重复行就是重复内容，不是要所有文件都重复才是。。

○ 需要打印重复内容并且出现该重复内容的文件名。

○ 第一题是不是用 hashset？但是那么多文件的话会存不下吧。。。

○ 是的，我朋友说可以把每一行的内容先哈希掉存进去

30. 一个数组，给两个数代表一个区间，让你反转这个区间内的数据，返回新的数组。需要考虑区间刚好跨过数组.输入是一个无序的数组，一个 i，一个 j。输出是反转后的数组。

31. 给一个 binary tree，deep copy the tree and add the level number to the value. eg. if the node is at level 2 and value is 3, the return tree should have value 5.

32. 给一个 array remove duplicates. 是 sorted 的

33. 523:<https://leetcode.com/problems/continuous-subarray-sum/>

34. 621:<https://leetcode.com/problems/task-scheduler/>

○ 简化版的，tasks 顺序不能变；

原题可以变顺序，所以需要个 PQ 来决定下一个做哪个 task，这题不用

--

哦这样！LC 的 solution 里最后一个直接算 idle slot 的，我想的是那个。如果这么做的话感觉一样吧？

对的，我觉得可行

---

想复杂了哈哈，顺序不能变 literally: A,\_,A,B,\_,B,\_,B -> 8

35. 56:<https://leetcode.com/problems/merge-intervals/>

○ <https://www.lintcode.com/problem/merge-two-sorted-interval-lists/description>

[2,4], [7,9]

[3,5], [11,13] ==> [2,5], [7,9], [11,13]

两个有序的 interval list, 求 intersection

○ follow up 是一个 list 过长怎么办 我就说 binary search

○ <https://leetcode.com/discuss/int...-two-interval-lists>

36. 791. Custom Sort String:<https://leetcode.com/problems/custom-sort-string/>

### 37. 105. Construct Binary Tree from Preorder and Inorder

Traversal:<https://leetcode.com/problems/construct-binary-tree-from-preorder-and-inorder-traversal/>

### 38. 443: <https://leetcode.com/problems/string-compression/> string compressing

○ follow up (代码能不能用 constant memory, 以及什么情况下不能还原压缩字符串等)

○ string compression (比如 'aaabbaa' -> 'a3b2a2') 并且问了一些非 coding 相关的 followup (比如说什么情况下 compress 之后没法还原)

○ 原字符串里有数字就没法还原, 比如说, 原字符串 'a2b2', 那得到的结果是 'a2b2', 可是再想重建时可能会生成 'aabb', 或者其他的 'a2bb' 等等

### 39. 767: <https://leetcode.com/problems/reorganize-string/>

40. bfs 他都一直在问, 你这逻辑是啥, 揪着我的 level order traversal 那行 code 问个死半天, 我把 bfs 简单叙述了一番, 结果他说: everyone knows how to run bfs. but can you explain. .

### 41. 76: <https://leetcode.com/problems/minimum-window-substring/> sliding window

### 42. 498: <https://leetcode.com/problems/diagonal-traverse/>

### 43. 636 : <https://leetcode.com/problems/exclusive-time-of-functions/>

○ follow up 是 event 会出错 要想到一些 corner case 然后抛出异常

○ follow up: 多线程会出现什么问题

### 44. 1123: <https://leetcode.com/problems/lowest-common-ancestor-of-deepest-leaves/>

### 45. 824: <https://leetcode.com/problems/goat-latin/>

46. 加上一个 vmstat 的题目。倒是没有考 vmstat 里面的每列都是什么意思, 只不过把它的 output 当作一个 file, parse 一下然后做一些 operation。讲真, 真的简单直接。如果说有什么建议的话, 就是一定, 一定要好好练一下 string manipulation 和 file IO

### 47. 139: <https://leetcode.com/problems/word-break/>

(139/636??)我用的 quick-select 递归的写法, 不过用 heap 应该也行, 第二道主要还是要沟通好, 其实面试官问法不一样, 本来是问 indexOf pattern in a string, 但是 pattern 可以是 anagram, 所以中间走了弯路, 如果一开始意识到是找 anagram 应该能把 sliding window 的解法写完

第二题想要的是 anagram 的 index, 找到就 return, 问的时候很像是写个 indexOf 就完了只是要包括 anagram 的情况。

### 48. 953 : <https://leetcode.com/problems/verifying-an-alien-dictionary/>

○ 给一个 wordlist 和一个自定义字母 order 的 list, 判断 wordlist 是否满足要求的 order, 也问了时空复杂度

○ 问了空间 optimize 方法 还有时间复杂度

### 49. 973: <https://leetcode.com/problems/k-closest-points-to-origin/>

### 50. 253: <https://leetcode.com/problems/meeting-rooms-ii/>

○ 这题最优的是按时间排

### 51. 给一个 sorted 的 array, 找到 target k 出现的次数。伞斯

(<https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/>)的解法, 但是在写码之前说错了时间复杂度, 说成了  $n \log n$  原因是因为两边都可能要运行。写

完后发现自己说错了，就纠正了是 logn。但是三姐貌似是想诈我，一直说你确定是 logn 吗，两边都要运哦，两边都要运哦，你再想想你想想。。。我问他是不是要 worst case，bestcase 和 ave case 分开看，她说不用。。。虽然搞得我有点虚，但是我最后还是坚持了 logn。她最后问我你的最终答案，我说 logn

52. input: array of integers 如果每个数字都是偶数个 就返回 True，如果有数字是奇数个 就返回 False。

○ 先用 hashmap 做。follow up 是问 space complexity 是  $O(1)$  的解法。第一题应该用 XOR 异或来实现全部数字异或，如果最后结果为 0 则 true 否则 false

○ bit manipulation

○ Sort?

53. 490: <https://leetcode.com/problems/the-maze/>

○ follow up shortest path ==> 505: <https://leetcode.com/problems/the-maze-ii/>

54. 347: <https://leetcode.com/problems/top-k-frequent-elements/>

○ 给了 min heap 的解法，之后还问了怎么 test

55. 350: <https://leetcode.com/problems/intersection-of-two-arrays-ii/>

○ 给两个 sorted array，输出两个 array 都包含的元素。

我说可以按照 merge array 的方法，用两个 moving pointer。没有让我实现。

○ follow up 1: 如果一个 array 很长，另一个很短，如何优化。

我说可以在长的 array 里面二分查找短的每一个元素。

○ follow up 2: 如果 array 里面有重复元素，重复的也要输出。比如

$a1=[1,2,2,2,3,4]$ ,  $a2=[2,2,3,3]$ , 需要输出  $[2,2,3]$ 。

我说需要保证每次 lo 停留在满足条件的最左边，之后从上次停留的地方开始接着搜索。

• 349: <https://leetcode.com/problems/intersection-of-two-arrays/>

56. 29: <https://leetcode.com/problems/divide-two-integers/>

• 稍微包装一下，并且还不要求负数的情况（面试官似乎也没有想到负数的情况，我问他他还有点惊讶，最后时间有多就作为 follow up 问了一下）

57. 124: <https://leetcode.com/problems/binary-tree-maximum-path-sum/>

58. <https://www.1point3acres.com/bbs/interview/facebook-software-engineer-548345.html>

• 有两个 List，分别代表两个人 Calendar 上面的空闲时间段。比如：

$A = [[1, 3], [5, 7]]$

$B = [[2, 4], [6, 8]]$

要求输出两人共同的空闲时间段。这题返回  $[[2,3], [6,7]]$  就好。

答：先记录下每个人的起始时间和结束时间，然后所有时间从左到右扫一遍，记录所有的“两人都空闲”的时间段即可。

• Follow-up 1: 如果要求共同空闲时间必须超过一定时间（比如，一个小时），怎么办？

答：加一个 if 判断即可。

• Follow-up 2: 如果假设[3, 4]和[4, 5]也有共同空闲时间[4, 4]（可以见一面），怎么办？

答：在每个时间点上，先考虑谁变得 available 了，处理一下，再考虑谁变得不 available 了。

• Follow-up 3: 如果有 20 个人，怎么办？

答：用 heap。

59. <https://www.1point3acres.com/bbs/interview/facebook-software-engineer-545515.html>

类似 file system

input = [/a/b/c,/a/b,/x/y]

output = [/a/b,/x/y]

input = [/a/b/c,/a/c]

output = [/a/b/c,/a/c]

input 是 list of file path，如果这个 path 的 prefix 和 input 里的某一个 file path 一样，就不需要把这个 path 放进 output 里。顺序是 unordered.

60. 426: <https://leetcode.com/problems/convert-binary-search-tree-to-sorted-doubly-linked-list/>

• 四二六 变体，普通的 binary tree，没有 sort，楼主用了 recursion 解法，interviewer 问的很细致，需要跑 case 解释代码

• 中间问了很多 case（楼主也主动提出了一些 corner case，比如 highly skewed，向左向右 skew，只有一个 node 等等）代码是否正确 和相关问题，比如 recursion 有啥问题没（回答了可能会 stack overflow，以及可以手工用 stack 来模拟递归的 implicit stack）

• 想问一下不是 bst 的话，convert 的 list 也不需要 sorted 了吗？只是把 binary tree convert 成 list？

是的，只需要 convert 就行了。全程都没提到 sort 不 sort，而且 sorted 还是 unsorted 和题目也没关系。

• followup 是反过来从 circular doubly linkedlist 转 binary tree，这个楼主解释了如何用 recursion 去做，中间卡住了一会儿又改用 iteration

§ 由 doubly linkedlist 转 binary tree 应该转出来的不唯一吧，请问下楼楼是怎么转到最初的那棵树的？选不同的点做 root 转出来的树就该不一样哦，就算 root 一样，下面的节点位置也可不一样。。

§ 主要是 head 是 tree 的 root 就行了，至于唯一性倒是没要求

• <http://www.zrzahid.com/transformflatten-binary-tree-tofrom-linked-list/>

61. 65: <https://leetcode.com/problems/valid-number/>

valid number 稍微简化的，没有逗号和指数处理之类的，但有正负和首位 0 之类的处理。明确说明考察 edge case 处理

# str: "12345" --- absolutely any string -> int32: 12345



其实一开始很纠结，这是道 hard 题 用 state machine 做法，有很多 states 要处理，

后来和 interviewer 不断沟通输入输出才确定不需要考虑很复杂的。

我自己给了一些例子比如：

1,233, 456 这个他说不用考虑这么复杂

0123 -> 123

-12 34 -> -1234

1a23 -> invalid （自定义输出指，楼主用的 Python 就输出了 None）

楼主解法是逐个反向 iterate 每个字符，转成 int 再乘 10 累加  
大概是这么写的：

```
def convert_number(s):
    no = 0
    for i, c in enumerate(s):
        try:
            c_i = int(c) # interviewer 说单个字符可以用 int 转， 如果不行， 其实可以
            # 用 ord(c) - ord('0')
        except:
            pass
        if c_i == 0:
            continue
        if i == 0:
            no += c_i
        else:
            no += c_i * 10 ** (i - 1)

    return -no if s[0] == '-' else no
```

62. 278: <https://leetcode.com/problems/first-bad-version/>

First Bad Version 变体， 看起来简单的二分查找， 其实考察点非常细， 而且页非常注重 testcase 和 communication

```
17 <-- good
18
19
20
21
22 <-- bad
```

# Given. Takes 11 hours to run.

def is\_good(svn\_version) -> bool:

first\_bad(good=3, bad=6) -> 5

```
is_good(3) -> True
is_good(4) -> True
is_good(5) -> False
is_good(6) -> False
```

```
# Implement this one
def first_bad(good, bad) -> int:
```

沟通真的很重要，确认输入输出含义，怎么转化成二分查找。其中最重要的是如何找 testcase 以及优化 (注意 call 一次 is\_good 要花费很长时间)

说他 nice 是因为写完了基本的 binary search 他会带着跑一些 case (楼主想先自己跑来着，他就直接开始了)，并且给出可能的优化点

楼主的做法：

```
def first_bad(good, bad) -> int:
    while good < bad:
```

```
        mid = (good + bad) // 2
```

```
        if is_good(mid):
```

```
            good = mid + 1
```

```
        else:
```

```
            bad = mid
```

# 注意以下两行，一开始放在了 mid 之前，被带着跑 testcase 发现问题，后来放这里了，稍微检查了一下 (时间不够了)

```
        if bad - good <= 1:
```

```
            return bad
```

```
        return good
```

下面是几个 testcase (其实最好自己能提出来，不够 interverer 好像也不是很介意)

我解释了奇偶数目的 case，不过 interverer 很关心 is\_good 的 call 次数，反复举例子可以优化

标 \* 表示 bad version

```
# *
```

```
# 3, 4, 5, 6
```

```
# *
```

```
# 3, 4, 5
```

# \*

# 3, 4, 5, 6

# \*

# 3, 4, 5

# \*

# 3, 4