

1

第一题

String to integer (利口-吧), followup: 如果不是十进制, 参数传一个 base (比如 `converter(String binaryString ("101001110001"), int base (2))`), 转换成根据那个 base 的 integer 值. 第二题 思思散(string compression)

第三题 给个 array, 对于 array 里每个 index 可以跳到 index 为 `index + arr[index]` 或者 `index - arr[index]`. 问能不能跳到值为 0 的 index 上

```
def jump_idx(arr):
    if not arr: return False
    elif len(arr)==1: return True if arr==[0] else False
    q = [0]
    visited = {0}
    while q:
        cur = q.pop()
        if arr[cur] == 0: return True
        if cur-arr[cur]>=0 and cur-arr[cur] not in visited:
            q.append(cur-arr[cur])
            visited.add(cur-arr[cur])
        if cur+arr[cur]<len(arr) and cur+arr[cur] not in visited:
            q.append(cur+arr[cur])
            visited.add(cur+arr[cur])
    return False
print(jump_idx([1,3,2,0]))
```

第四题. 长度为 N 的 array 里的元素都是 0-N, 求出任意一个 duplicate

1.construct bst 这类题需要 compile run 吗? - 不需要 只要口头过一遍就好

2. most frequent words

3. Write a func $Y = F(X)$ where x is an array of floats and Y is an array of integers with the following conditions: $\text{Sum}(Y)$ is equal to the round of $\text{Sum}(X)$, Y_n is either floor or ceiling of X_n , $\text{Sum}(\text{Abs}(X_n - Y_n))$ is the minimum (dfs)

```
import math
def float2int(x):
    sum_x = round(sum(x)) #round 6.5 = 6
    min_diff = float("inf")
    res_y=[]
    def dfs(i, cur_sum, cur_diff, tmp_y):
        nonlocal sum_x, min_diff, res_y
```

```

if i==len(x) and cur_sum==sum_x:
    if cur_diff<min_diff:
        min_diff=cur_diff; res_y=tmp_y
    return
if cur_sum>=sum_x or i>=len(x): return
y = math.floor(x[i])
bfs(i+1, cur_sum+y, cur_diff+abs(y-x[i]),tmp_y+[y])
y = math.ceil(x[i])
bfs(i+1, cur_sum+y, cur_diff+abs(y-x[i]),tmp_y+[y])
return
dfs(0,0,0,[])
return min_diff, res_y
print(float2int([1,1.99,1.49,1,1]))

```

4. string compression 比你早一周面的巨硬招聘会，有三轮是一样的。。。

微软招聘会 9/18

招聘会 4 轮， 1. 断句， 一个长字符串， 按要求满 20 length 就换行， 完整单词换行
 2. boundary of binary tree
 3. most frequent words in List
 4. StringBuffer String 字符串操作

Microsoft 9.22 OneEng hiring event. 4 轮视频面试。

第一轮： bq 做过的 projects， 写 LinkedList data structure

第二轮： MinStack 蠡口易武武

第三轮： 一个 n-ary 的 tree 有 child， sibling 和 parent， <https://www.geeksforgeeks.org/cr ... ling-representation>， 找到 Next Node

今天刚刚面完的 virtual onsite 第一轮：

<https://leetcode.com/problems/number-of-distinct-islands/>

第二轮：

1. 线程进程区别 2. UDP vs. TCP 3. 乐观锁 vs. 悲观锁 4. 手写最小堆(minPriorityQueue)

第三轮：

<https://leetcode.com/problems/string-compression/>

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

巨婴视频上门

第一轮：利口 二三就+ 以三起三

第二轮：bq+俩字符串相加，follow-up：字符串特别长怎么搞

就是两个 string 代表的数字相加，string 从高位到低位，比如 “12345” + “2345”
=12345+2345

```
def add(s1,s2):
    i1, i2 = len(s1)-1, len(s2)-1
    res,i,tmp = "",0,0
    while i1>=0 and i2>=0:
        _sum = int(s1[i1])+int(s2[i2])
        if _sum+tmp<10:
            res = str(_sum+tmp) + res; tmp = 0
        else:
            res = str(_sum+tmp-10) + res;tmp = 1
        i1-=1; i2-=1
    if tmp == 1:
        if i1>=0: res = str(int(s1[:i1+1])+1)+res
        elif i2>=0: res = str(int(s2[:i2+1])+1)+res
    else:
        if i1>=0: res = s1[:i1+1]+res
        elif i2>=0: res = s2[:i2+1]+res
    return res
```

第三轮：bq+ 字符串判回文 + 字符串中找 size-k 的回文

```
def countAllPalindrome(S): #Time O(N^2), Space O(1)
    N = len(S)
    res = 0
    for center in range(2*N - 1):
        left = center // 2
        right = left + center % 2
        while left >= 0 and right < N and S[left] == S[right]:
            res += 1; left -= 1; right += 1
    return res
```

VO :

R1: LRU + fu how to make it thread safe + task scheduler (topological sort)

R2: 1. given two integer x, y, each step you can do 1) $x * 2$ or 2) $x - 1$, how many steps at least does it take to convert x to y? 2. edit distance

<https://www.geeksforgeeks.org/minimum-number-operation-required-convert-number-x-y/>

```

def minOperations(x, y):
    q = [[x, 0]]
    visit = {x}
    while q:
        t = q.pop(0)
        print(q)
        if t[0] == y: return t[1]
        if t[0] * 2 == y or t[0] - 1 == y: return t[1] + 1
        if t[0] * 2 not in visit:
            q.append([t[0] * 2, t[1] + 1]); visit.add(t[0] * 2)
        if t[0] - 1 >= 0 and t[0] - 1 not in visit:
            q.append([t[0] - 1, t[1] + 1]); visit.add(t[0] - 1)

```

R3: design garbage collection 主要口头说，他的考点是让你用 bfs/dfs 从已知的 reference 找到所有 referenced objects。design web crawler。

R4 : report manager 主要在介绍组里情况 + answer questions

R5 : hiring manager process 和 thread 相关知识，how do threads in the same process / threads in different processes/ process in different boxes...

communicate? 还做了一道算法 但是记不清了。最后 10min 问问题

巨硬招聘会现场表演

第一轮 bq 第二轮 力扣 腰吴 第三轮 力扣 吴耳酒 简化版 第四轮 力扣 遛散酒 简化版

3sum: <https://leetcode.com/problems/3sum/solution/>

```

def threeSum(self, nums):
    res = []
    nums = sorted(nums)
    for i in range(len(nums) - 2):
        if i > 0 and nums[i] == nums[i-1]: continue
        l, r = i + 1, len(nums) - 1
        while l < r:
            s = nums[i] + nums[l] + nums[r]
            if s == 0:
                res.append([nums[i], nums[l], nums[r]])
                l += 1; r -= 1
                while l < r and nums[l] == nums[l - 1]: l += 1
                while l < r and nums[r] == nums[r + 1]: r -= 1
            elif s < 0: l += 1
            else: r -= 1
    return res

```

Decode Ways II <https://leetcode.com/problems/decode-ways-ii/>

```

class Solution(object):
    def numDecodings(self, s: str) -> int:
        if s[0] == '0': return 0
        n = len(s)
        dp = [0] * n
        dp[0] = 9 if s[0] == '*' else 1
        for i in range(1, n):
            case1 = (9 if s[i] == '*' else 1) # '*' or '1*'
            case2 = (6 if s[i] == '*' else (1 if s[i] < '7' else 0)) # '2*'
            pre1 = dp[i-1]
            pre2 = dp[i-2] if i-2 >= 0 else 1
            if s[i] != '0': dp[i] = pre1 * case1
            if s[i-1] == '1': dp[i] += pre2 * case1
            elif s[i-1] == '2': dp[i] += pre2 * case2
            elif s[i-1] == '*': dp[i] += pre2 * (case1+case2)
        return dp[n-1] % (10**9 + 7)

```

第一轮和第二轮题都能在 leetcode 上找到，意思流和思八九。第三轮系统设计，需要设计一个下载系统，第四轮是一个和 Trie 概念相关的题，medium 难度，知道 Trie 相关概念就差不多。

489. Robot Room Cleaner <https://leetcode.com/problems/robot-room-cleaner/>

```

def cleanRoom(self, robot): #旋转 Backtracking
    def go_back():
        robot.turnRight();robot.turnRight();robot.move();robot.turnRight();robot.turnRight()
    def backtrack(cell=(0, 0), d=0):
        visited.add(cell)
        robot.clean()
        # going clockwise : 0: 'up', 1: 'right', 2: 'down', 3: 'left'
        for i in range(4):
            new_d = (d + i) % 4
            new_cell = (cell[0] + directions[new_d][0], cell[1] + directions[new_d][1])
            if not new_cell in visited and robot.move():
                backtrack(new_cell, new_d)
            go_back()
            robot.turnRight() # turn the robot following chosen direction : clockwise
        directions = [(-1, 0), (0, 1), (1, 0), (0, -1)]
        visited = set()
        backtrack()

```

微软 Azure 数据库组面经

刚结束的微软 Azure 的一个数据库组的面经，一共五轮，题目都不难，而且面试的感觉总体也挺好。

第一轮：给了一个一小时的 PhD presentation，把 PhD 期间做的工作介绍了一下。

第二轮：两个 coding，第一个是 merge sort linked list，第二个是 read4 (FB 的高频题)

157. Read4 <https://leetcode.com/problems/read-n-characters-given-read4/>

```
def read(self, buf, n) -> int:
    copied_chars = 0
    read_chars = 4
    buf4 = [""] * 4
    while copied_chars < n and read_chars == 4:
        read_chars = read4(buf4)
        for i in range(read_chars):
            if copied_chars == n:
                return copied_chars
            buf[copied_chars] = buf4[i]
            copied_chars += 1
    return copied_chars
```

第三轮：一个 design/coding，实现一个 delayed scheduler 可以在将来执行一些 task，支持多线程和取消一个 task

第四轮：一个 coding，给了一个用 linked list 实现的 stack，让用 compareAndSwap 让它支持多线程。然后问了一个 design 的问题，怎么设计一个 data store 支持 expire 操作（某个 key 可以在一段时间后失效）

第五轮：一个大老板的面试，问了一些 BQ 还有一个 design 的问题，怎么设计一个系统去计算 term frequency (跟 inverted index 相关)。

巨硬招聘事件面经

第一轮 给一句话和一段话 判断能不能由一段话里的字母重构那一句话

本质就是一个短 string 一个长 string 问长 string 的字母能不能组成短 string

第二轮 印度姐姐 酒罢留 + bq

986. Interval List Intersections

```
def intervalIntersection(A, B):
    if not A or not B: return []
    i, j, res = 0, 0, []
    while i < len(A) and j < len(B):
        max_s, min_e = max(A[i][0], B[j][0]), min(A[i][1], B[j][1])
        if max_s <= min_e: res.append([max_s, min_e])
        if j < len(B) - 1 and i < len(A) and B[j][1] <= A[i][1]: j += 1
        else: i += 1
    return res
```

第三轮 sort array 我讲了讲 merge sort 正准备上手写的时候说不用写了, 问我如果有很多个大 file 每个 file 里都是 sort 好的数 该如何 merge 这些 file 本质就是 merge k sorted list

MergeSort

```
def mergesort(seq):
    if len(seq) <= 1: return seq
    mid = len(seq) / 2 # 将列表分成更小的两个列表
```

```

left, right = mergesort(seq[:mid]), mergesort(seq[mid:])
return merge(left, right)

def merge(left, right):
    result = []
    i, j = 0, 0
    while i < len(left) and j < len(right):
        if left[i] <= right[j]: result.append(left[i]); i += 1
        else: result.append(right[j]); j += 1
    result += left[i:]
    result += right[j:]
    return result

```

23. Merge k Sorted Lists

```

from Queue import PriorityQueue
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val; self.next = next
class Solution(object):
    def mergeKLists(self, lists):
        head = point = ListNode(0)
        q = PriorityQueue()
        for l in lists:
            if l: q.put((l.val, l))
        while not q.empty():
            val, node = q.get()
            point.next = ListNode(val)
            point, node = point.next, node.next
            if node: q.put((node.val, node))
        return head.next

```

但是我一遇到 file 的读写就发怵 不知道咋写 我想用 array 或者 list 来表示大叔说不行 后来大叔说你就 file.read() 就行，每次 file.read() 了以后就会自动挪到下一个数

第四轮 应用题，给 a list of vote，选出 most voted candidate 本质就是 top k
follow up 是如果 vote 里包含 candidate 的 party 信息，要求出不同的 party 的 top k 该怎么办 答按 party 来 sort

347. Top K Frequent Elements

```

def topKFrequent2(self, nums, k): #O(n)
    bucket = [[] for _ in range(len(nums) + 1)] # of freq
    count = Counter(nums).items()
    for num, freq in count: bucket[freq].append(num)
    flat_list = [item for freqs in bucket for item in freqs]
    return flat_list[::-1][:k]
def topKFrequent(self, nums, k): #O(n)-O(nlgn)

```

```
count = Counter(nums)
x = sorted(count.items(), key=lambda kv: -kv[1])
return [xx[0] for xx in x[:k]]
```

第一轮 code + [系统设计](#) + BQ

code 是 find number of Palindrome string 要求长度大于 2. 系统设计 简要设计 one drive

第二轮 儿伞吧 可以用除法 但是不推荐 因为要是有一个 0 就酸爽了

第三题 好像用二分法求个啥

第四题 应用题

一个 CSV reader 读取 csv 文件实现 下面几个接口

boolean hasNext(); String next(); String get(int columnIndex); String get(String columnName);

题目很简答主要 exception 处理

巨硬 event 挂经

1. bq 最有挑战的

然后两道题 decode string 和 删掉 linkedlist 里面重复的 node

然后问了一下 如何设计停车场的应用，就讲了一下 user case 就到时间了

2. 直接做题

一道题 先是 reverse a word 然后 给你一个 string reverse 里面的每个 word，考虑空格啊之类的

然后设计一个找到附近朋友的应用

3. 第一题 设计一个只能被写一次的类，要考虑线程安全，然后加锁 第二题 原题伞咦

31. Next Permutation

```
def nextPermutation(nums): #O(n) - O(1)
    if not nums: return []
    i = len(nums) - 1 #1. 从后往前找到第一个降低的数
    while i >= 1 and nums[i] <= nums[i - 1]: i -= 1
    if i == 0: nums.reverse(); return
    k = i #2. 再往后找第一个大的数 · swap, 再 reverse 后面这一团
    while k < len(nums) and nums[k] > nums[i - 1]: k += 1
    nums[k - 1], nums[i - 1] = nums[i - 1], nums[k - 1]
    nums[i:len(nums)] = nums[i:len(nums)][::-1]
```

4. 是跟 director 聊 是一个从毕业就呆在他家的三哥

大部分时间都在瞎聊 bq 啊 简历啊

中途说我们来做个题 让你 dfs 找一个图的 node 就完了，我开始想成了 bfs 结果写的时候发现了就立马改过来了 写完之后又开始聊天 还给你邮件让你之后有问题问他

招聘事件面经

第一轮，大叔，BQ+简历+OOP+一个小小的[系统设计](#)，OOP 是交通工具

第二轮，大叔，BQ+简历+系统设计，设计一个推荐系统+分享照片系统的合集系统 = =

第三轮，小哥，BQ+简历，一道非常简单的算法题，维护一个字典，然后写一个 function 不停的 update 这个字典，估计是前几轮跟面试官吹 JS 用的不错，小哥上来就问，要不你用 JS 写吧。。愣是思考了一会 JS 的 hashmap 长什么样。写完 code 写 test，然后 debug 和 run。中间 compile fail 了几次都是些小 syntax 问题略丢人，还是太紧张。

第四轮，BQ 2 分钟然后 coding，链表有 randompointer，做 deep copy 秒，接着一道链表找环秒，一些基本的 follow up 结束。

138. Deep Copy List with Random Pointer

```
class Solution(object):
    def __init__(self):
        self.visitedHash = {} # key: old nodes value: new nodes
    def copyRandomList(self, head):
        if head == None: return None
        if head in self.visitedHash: return self.visitedHash[head]
        node = Node(head.val, None, None)
        self.visitedHash[head] = node
        node.next = self.copyRandomList(head.next)
        node.random = self.copyRandomList(head.random)
        return node
```

微软 8/7 hiring event IRIS 部门

2. 亚裔大叔，问了点 BQ，明显看出来完全是照着念的题目。。就是非常机械的一问一答。。。然后是一道 string 的题目，reverse string 里的每个单词。面到一半说我们再来个[系统设计](#)，设计一个可以找到周围朋友的系统，没剩多久了，随便聊聊结束

557. Reverse Words in a String III

```
start, res = 0, ""
for i in range(len(s)):
    if s[i] == " ":
        if i > start: res += s[start:i][::-1]
        res += " "
        start = i + 1
res += s[start:][::-1]
print(res)
```

3. 上来一个很简单的设计，一个变量只能被 set 一次，然后开始了可怕的多线程。。。

然后是一个原题伞移，不要求写代码，说是只想知道你的 thought process。

4. 大部分都是 BQ，然后随手写了个简单的 dfs，就是在图里找一个特定的 node

5. Bq, 原题救移

91. Decode Ways

```
def numDecodings(self, s):
    if not s: return 0
    dp = [0 for _ in range(len(s) + 1)]
    dp[0] = 1
    dp[1] = 0 if s[0] == '0' else 1
    for i in range(2, len(dp)):
        if s[i - 1] != '0': dp[i] += dp[i - 1]
        two_digit = int(s[i - 2: i])
        if two_digit >= 10 and two_digit <= 26: dp[i] += dp[i - 2]
    return dp[len(s)]
```

微软挂经

1. 和最大的 subarray, 求这个和是啥，follow up 求返回和最大的一个 subarray

53. Maximum Subarray

```
def maxSubArray(self, nums):
    n = len(nums)
    max_sum = nums[0]
    # tmp, res=[nums[0]], [nums[0]] #找 path
    for i in range(1, n):
        if nums[i - 1] > 0:
            # tmp.append(nums[i])
            nums[i] += nums[i - 1]
        # else: tmp=[nums[i]]
        if nums[i] > max_sum: max_sum = nums[i]; # res=tmp[:]
    return max_sum
```

2. 不是一个算法题，给一个 String，然后按照他给定的要求 parse 成 object，比如 "12345, dsfjbhweew, 1,2,3,4" 12345 作为返回 object 的一个 attribute，dsfjbhweew 作为另外一个 attribute，1,2,3,4 作为一个 list，没明白这有什么意义

3. 判断一个 String 是不是一个 valid ip address, 只需要返回 true false，第二题 [刷题网](#) 而无尔，我说了一下思路，三哥面试官表示你已经知道怎么写了那就别写了，黑人问号

4. 忘记是啥了，但是也是一个刷题网 easy

热腾腾的阿祖儿 storage

1. 一个不知长度的数组，可调用 getElementAt(index), 写 doesElementExist(Element e), sorted, 无重复 fu: 一个 queue interface. 有四个 api: enqueue, dequeue, isFull, isEmpty 都不 thread safe. 写个

treadSafeQueue wrapper class

2. merge two binary search tree, 要求 space logn (inorder 变 pointer 成 linked list)

617. Merge Two Binary Trees

```
def mergeTrees(t1, t2):
    if not t1:
        return t2
    elif not t2:
        return t1
    t1.val += t2.val
    t1.left = mergeTrees(t1.left, t2.left)
    t1.right = mergeTrees(t1.right, t2.right)
    return t1
```

617. (Print) Merge Two Binary Search Trees

```
class newNode:
    def __init__(self, val: int):
        self.val = val; self.left = None; self.right = None
# def inorder(root):
#     if root: inorder(root.left); print(root.data, end=" "); inorder(root.right)
def merge(root1, root2):
    stack1, stack2 = [], []
    tree1, tree2 = root1, root2
    result = []
    while (tree1 or tree2 or stack1 or stack2):
        while (tree1):
            stack1.append(tree1); tree1 = tree1.left
        while (tree2):
            stack2.append(tree2); tree2 = tree2.left
        if (not stack2) or (stack1 and stack1[-1].val <= stack2[-1].val):
            tree1 = stack1.pop()
            result.append(tree1.val)
            tree1 = tree1.right
        else:
            tree2 = stack2.pop()
            result.append(tree2.val)
            tree2 = tree2.right
    return result
root1 = newNode(3); root1.left = newNode(1); root1.right = newNode(5)
root2 = newNode(4); root2.left = newNode(2); root2.right = newNode(6)

merge(root1, root2)
```

3. 一个 file 里有 0-31999 这些数字, unsorted, 可调用 read() 每次返回一个数字, 可调用 write(File dest) 每次写一个数字到 destination file, 要求写一个方法 read 之后排好序写到新的 file 里, 要求 time o(n), 怎么优化 space. (bucket sort) fu: 很神奇跟第一轮是一个题 不过问的角度不太一样

4. 八五二 fu: 有 duplicate fu: arr 很长 很多 duplicate 怎么 parallel

软家虚拟在线

1. Round 1: 设计和实现 binary search tree, 需要设计 tree structure 和实现 search 功能, 一些 BQ question
2. Round 2: 找出 the longest contiguous subarray (contiguous 可以是递增的, 也可以是递减的), 设计一个 online game 的 player match 功能 (比如你玩 FIFA, 选择一个 match 后, 帮你选择适合的在线玩家一起打比赛)
3. Round 3: BQ question

longest increasing (contiguous) subarray

```
def lenOfLongIncSubArr(arr, n):  
    m, l = 1, 1  
    for i in range(1, n):  
        if (arr[i] > arr[i - 1]):  
            l = l + 1  
        else:  
            m = max(m, l); l = 1  
    m = max(m, l)  
    return m
```

微软之越来越像 amazon 的虚拟 onsite

- 题目 : 1. Meeting room ;
2. 树那个题, 加个 next node, 左右连一块儿那个 ;

Connect nodes on same level

```
class newnode:  
    def __init__(self, data):  
        self.data = data; self.left = None; self.right = None; self.nextRight = None  
  
def getNextRight(p):  
    temp = p.nextRight  
    while (temp != None):  
        if (temp.left != None): return temp.left  
        if (temp.right != None): return temp.right  
        temp = temp.nextRight  
    return None  
  
def connect(p):  
    temp = None  
    if not p: return  
    p.nextRight = None # Set nextRight for root  
    while (p != None): # set nextRight of all levels  
        q = p  
        while (q != None): # Connect all children and others at same level  
            if (q.left):  
                if (q.right): q.left.nextRight = q.right  
                else: q.left.nextRight = getNextRight(q)  
            if (q.right): q.right.nextRight = getNextRight(q)
```

```

    q = q.nextRight
    if p.left: p = p.left
    elif p.right: p = p.right
    else: p = getNextRight(p)

```

3. 好像是说明 decode 一个 url , 重点是 convert hex to ascii , (不清楚到底要考个什么鬼 , 直接用 api 搞定了。)

hex to ascii

```

bytearray.fromhex("7061756c").decode()

```

巨硬 Onsite 面经 Onsite , 365 org

第一轮 : skip level manager : BQ , 设计 ingest data 的 system

第二轮 : hiring manager : BQ , 希望减低 ops 的开支和提高 ops , 讨论解决方案

第三轮 : 做题 : 设计 linkedlist , iterate linkedlist , deep copy linkedlist。聊计算机基础知识 : reference , deep copy , hashmap 等

第四轮 : 做题 : 利口 - 期久

79. Word Search

```

def exist(self, board: List[List[str]], word: str) -> bool:

```

```

    def search(word, i, j, match_idx):

```

```

        if match_idx == len(word) - 1: return True

```

```

        for dx, dy in directions:

```

```

            x = i + dx

```

```

            y = j + dy

```

```

            if 0 <= x < len(board) and 0 <= y < len(board[0]) and visited[x][y] == False and board[x][y] == word[
                match_idx + 1]:

```

```

                visited[x][y] = True

```

```

                if search(word, x, y, match_idx + 1):

```

```

                    return True

```

```

                visited[x][y] = False

```

```

m, n = len(board), len(board[0])

```

```

directions = ((1, 0), (0, 1), (-1, 0), (0, -1))

```

```

visited = [[False for _ in range(n)] for _ in range(m)]

```

```

for i in range(m):

```

```

    for j in range(n):

```

```

        if board[i][j] == word[0]:

```

```

            visited[i][j] = True

```

```

            match_idx = 0

```

```

            if search(word, i, j, match_idx): return True

```

```

            visited[i][j] = False

```

```

return False

```

2. 此轮较简单 : 两道键蛋的里扣 , 武武旗 和 么灵么

101. Symmetric Tree

```
def isSymmetric(root):
    def isMirror(t1, t2):
        if not t1 and not t2: return True
        if not t1 or not t2: return False
        return (t1.val == t2.val) and isMirror(t1.right, t2.left) and isMirror(t1.left, t2.right)
    return isMirror(root, root)
```

迷宫问题，此轮有 shadow -> 给起点终点和障碍物，问能否从到达终点。bfs/dfs 迅速解决，follow up 和复杂度+ml 相关，问如果 block 是 sparse 的能否降低复杂度。懵逼未答上来
第二轮码题，刷题网 石，一开始是个简单版本的，只有 点，后来上完整版 有星号。

10. Regular Expression Matching

```
def isMatch(self, text, pattern):
    dp = [[False] * (len(pattern) + 1) for _ in range(len(text) + 1)]
    dp[-1][-1] = True
    for i in range(len(text), -1, -1):
        for j in range(len(pattern) - 1, -1, -1):
            match = False
            if i < len(text) and pattern[j] in {text[i], '.'}: match = True
            if j + 1 < len(pattern) and pattern[j + 1] == '*':
                dp[i][j] = dp[i][j + 2] or (match and dp[i + 1][j])
            else:
                dp[i][j] = match and dp[i + 1][j + 1]
    return dp[0][0]
```

软家新鲜面经

三轮 1. 调试 codility 1015 分钟，尝试了 codility，打电话，最终用 teams 语音通话，codility share code。电话打通直接做题，ctci 189 那本书药零点Ⅷ的变形

2. 调试 codility 10 分钟（每个面试官都不会用？？？），最终 teams 视频，codility share code。BQ 10min，题目是 big file 里的每一条数据都不太正确，提供了一个 API 可以修复这个错误。要求修复每一条数据里的错误，并且对每一个 key，API 只能被调用一次。

3. 调试 codility 10 分钟（真的每个面试官都不会用！！），最终 teams 视频，codility share code。BQ 10min，题目是 ctci 189 要领点 IV。

这一轮面完面试官问我今天接下来什么安排，我说后面还有一轮面试。他告诉我从他看到 schedule 上，我没有第四轮面试。

巨硬 vo 跪经

加电面一共面了 5 轮 3 轮 binary search 2 轮 bst 电面是 伞司

第一轮面的是给一个 sorted array 只有一个 api getIndex(x) 让你招到 某个 element 是否存在在这个 array 里 不知道 boundary

第二轮是 尔把无

285. Inorder Successor in BST (Next Bigger)

```
def inorderSuccessor(root, p):
    stack, inorder = [], float('-inf')
    while stack or root:
        while root:
            stack.append(root)
            root = root.left
        root = stack.pop()
        if inorder == p.val: return root
        inorder = root.val
        root = root.right
    return None
```

第三轮是 把刘伞 写到一半和我说你不用写了我们看下一题

863. All Nodes Distance K in Binary Tree

```
def distanceK(self, root, target, K):
    def dfs(node, pre=None):
        if node: node.pre = pre; dfs(node.left, node); dfs(node.right, node)
    dfs(root)
    queue = collections.deque([(target, 0)])
    seen = {target}
    while queue:
        if queue[0][1] == K:
            return [node.val for node, d in queue]
        node, d = queue.popleft()
        for nei in (node.left, node.right, node.pre):
            if nei and nei not in seen:
                seen.add(nei); queue.append((nei, d + 1))
    return []
```

问我怎么 download 一个大文件 到 local file 里 给我了一个 readChunk 的 api 让我写
DownloadFile(remote, local)

第四轮 找一个先递增后递减 array 里的 peak

Q: 没有边界怎么 binarysearch 确定上界? 是从 $r = \max \text{int}$ 开始找还是需要先大步找到 out of index 再 binary search 找到边界点?

A: 可以用 exponential search · 从 step size = 1 开始 · 每次将 step size 翻倍。例如从 0, 1, 3, 7, 15 上的元素搜。

7.17 Azure security VO 挂经

1. 18 年经验的 principle manager 很和蔼 20 分钟 BQ 一题 Kth largest in a tree 最后还有 10 分钟, 乱聊 问道了 OAuth2.0 中 token reuse 的问题 (一直获取新 token, 没有 reuse, 如何 detect)

230. Kth Smallest Element in a BST

```
def kthSmallest(self, root, k):
    stack = []
    while True:
        while root:
            stack.append(root); root = root.left
        root = stack.pop(); k -= 1
        if not k: return root.val
        root = root.right
```

2. 15 年经验的 principle manager 很和蔼 介绍组，20 分钟 BQ，can form palindrome + number of unique palindrome 267. Palindrome Permutation II

3. 18 年经验的 principle manager 很和蔼 这轮面的不好：先 BQ，然后问了 API version 的问题（没用过，没答上来）现场表演 solve sudoku 没让跑 test，问我如何给这个代码做 cr，当时非常懵逼 现在回想觉得他应该是在问 SOLID principle。还有问我们 vend library，那怎么处理不同 consumer 的 log 设置不同 -这里再度非常懵逼，因为不知道他在问什么。。。他有提示我说有些 log 到 console 有些 log 到 file，怎么处理呢？最后他说传入 logger 就可以了，我才反应过来，说这是 dependency injection。。。。

4. 22 年经验的 partner manager，比较严肃 这轮面的不好：以为是 [system design](#) 结果全程 OAuth 2.0 的 domain knowledge 大讨论。主要讨论 token 被盗用的情况。我非常懵逼，因为我们假设 TLS 是安全的不需要我们去管..... 我磕磕绊绊给了一个双重 asymmetric encrypt 的办法把 token 也加密，在他提示下给出另外一个解法是把 token 对应的 ip 也加在 token 上... 感觉没答出他想要的，最后他跟我说是 OAuth 2.0 的 proof of possession。没用过，真的不知道。最后还把我当 TPM 问，说 2 个 security protocol，我就是要用旧的不肯用新的，你怎么来劝我.....

阿祖儿安全组 VO

阿祖儿的一个安全组的 VO。第一轮[刷题网](#)八八+九二

92. Reverse Linked List II

```
def reverseBetween(self, head, m, n):
    if not head: return None
    left, right = head, head
    stop = False
    def recurseAndReverse(right, m, n):
        nonlocal left, stop
        if n == 1: return
        right = right.next
        if m > 1: left = left.next
        recurseAndReverse(right, m - 1, n - 1)
        if left == right or right.next == left: stop = True
        if not stop:
            left.val, right.val = right.val, left.val
            left = left.next
```



```
recurseAndReverse(right, m, n)
return head
```

第二轮是个一轮 [system design](#) , 先过了一轮简历。问题问的是怎么设计一个 group chat。这个 chat 可以分享视频和文字。

第三轮是一个 system design , 也是先过了一轮简历。从一个 Apple music 的 interface 开始怎么设计后端的不同的方块。重点聊了下载怎么方便, 还有 playlist md 怎么存。。

第四轮问的是给一个 String, 求 number of permutation, 面试官给了一个公式, 一开始有点愣, 但过了几分钟才意识到套公式就好。

第五轮和一个相当于 senior manager, 聊 system design 问一个例子如何设计一个钥匙管理的服务能保证用一个账户密码能同时用在几个银行里, 这轮交流有点困难。。

巨硬 VO

1. 三姐 各种 push 体验很不好

<https://www.geeksforgeeks.org/find-a-missing-number/> 她是想让我用方法三。

2. 一个简单的雇员和雇主的递归题目, 我理解错题目了, 还整了半天拓扑结构 哎

3. 自己出的一个现实问题, 但是其实就是一个 permutation

4. 而霸气 要求 $O(1)$ 我用了 linkedlist 找环的思路, 面试官让我证明这个思路是对的

微软 5 月底虚拟面试

第一轮, 一个中等难度的链表题

第二轮, 求数组乘积

第三轮, 经典小岛题, 问一个 Matrix 里有几个小岛

第四题, 给一个 API function, getChars, 每次返回 X 个字符, 写一个 function 来抓 N 个字符。

巨硬 阿祖儿 招聘事件

1. 东欧小哥。上来 BQ, 然后出题。(1) 给出两个点的坐标算两个点之前的距离。(2) 给 N 个点的多边形, 算周长。(3) 给 N 个点的多边形, 算面积 (可以是凹或者凸的)。(4) 具体不大记得了是和算斜率相关的。

2. 白人小哥, 上来 BQ。然后出题。 [刷题网](#) 珥陵溜和迩武。

3. 白人经理。BQ 加题目, 一个数组, 数组里面有正数和负数, 正数和负数的数量差在一个以内 (1. 正数比负数多一个 2. 正数负数相同 3. 负数比正数多一个)。要求按照正负数相间的形式重新排列数组。

4. 东欧老大爷。BQ 家题目。 给一个字符串要求输出所有大小写的组合如果出现其他一些符号要忽略。 要求只能用 iteration 的办法。 Sample: as\$: as\$, As\$, aS\$, AS\$

reverse Nodes in k-Group, 第三题需要 in place 吧, $O(n)$ space. 还有, 正数组和负数组的相对 order, 要保留么? 比如说, 1 2 3 -1 -2 -3, 成为, 1, -1, 2, -2, 3, -3, 还是说, 1, -2, 3, -1, 2, -3 也是可以的呢? 我感觉要 2 次 loop 才做得出来

巨硬面试 求大米看面经呀!

Merge LinkedList (Recursion + Iteration 两种方法)

Delete a LinkedList node with target value

Reverse String

<https://www.1point3acres.com/bbs/thread-670374-1-1.html>

因为有其他 offer, 一个 Azure team 直接说可以跳过电面, 直接 onsite。

第一轮: meeting rooms 2? bucket sort, 我记得好像是把它拆分成 event 形式更像?

你开 201 个空间, 然后去扫原数组: [1960, 2020], [1988, 2099] 这样, 看到 1960 是第一个元素你知道这是 birth, 那就在开的空间 [1960 - 1900] 这个位置 + 1, 看到 2020, 你知道这是第二个元素, 那就是死亡, 你在 [2020 - 1900 + 1] 这个位置 - 1, 以此类推。

最后扫一遍 201 个空间的数组, 一个 global max, 一个 running max, 201 个空间里面存的, 正数代表最后生的大于死的, 负数代表死的大于生的, 直接都加在 running max 里面, 在跟 global max 比较。不能说是完全的 bucket sort 吧, 有点像用 map 去存生死个数。不过当时我解释的时候, 是把每个年份当成一个 bucket, 你在记录相对应位置的死亡和生的数量时候, 相当于在 sort 年份。大概就是这么个意思

印度人, 不开摄像头, 不知道年龄。题目: Give you list of birth and death year, calculate max of population. 经过多轮讨论, 就是想求一个最大存活的 population 的人数。Input: {{1966, 2020}, {1988, 2099}}, 代表第一个人 1966 birth, 2020 年 death。第二个人 1988 年 birth, 2099 年 death, 那这样在 1988 到 2020 这个阶段就有两个人活着, 那这个例子返回 2. 出生和死亡年在 1900 - 2100 这个区间。并且死亡的人, 需要在第二年才算, 就是 2020 时候, 人数还是 2, 到 2021, 第一个人算死了, 人数变成 1. 思路: 先说了暴力方法。然后说了把 [1966, 2020] 拆成 [1966, birth], [2020, death] 这样 event 形式, 然后根据年排序后, 可以扫这个排序的数组, 算出最大 alive 的人数。面试官提醒, 你还没有用到区间, 想想有啥更快的么? 然后想到 count/bucket sort, 不用拆成 event, 直接根据这个区间, 开空间, 然后进行 bucket sort 就行了。面试官表示满意, 让我写 code, 最后写完,

还剩二十分钟。面试官说我没什么问题了，但是也没让我问问题，直接就挂了。。。。我去了。。。。雷的我外焦里嫩。。。这是干啥呢，这么着急。

这题看着很熟悉，一下想不起来哪里见过，然后发现是 facebook 的样例题。如果你面 facebook，HR 会给你发个长邮件教你怎么面试，这个题被当成样本来讲。private 的 video，不敢乱放到网络上。

第二轮：

印度大叔，有工作经验 14 年了，现在是 HM。问了一道：input: [4, 2, 6, 8], output: [6, 6, 8, -1]. 就是去找比当前元素大并且在右边最近的元素。对于 4，它右边比 4 大最近的就是 6，对于 2，也是 6，对于 6，是 8，对于 8，没有右边，返回-1. 最后一个元素永远返回-1. 思路：最开始说了暴力解法，然后面试官说优化下。用 Stack 存 index，保持一个单调递减，遇到比栈顶大的，就出栈，并且当前出栈的 index 位置设置成当前元素。扫一遍数组就好了。最后栈里面会有剩余元素，所有剩余元素，都设置成-1。

follow up：如果这个数组是循环数组怎么办？copy 数组

follow up: 不想开那么多空间怎么办？

第三轮：

Q：长度为 K 的 subsequence 中最小的那个 感觉用一个 stack 可以解决？

A：长度最小，并且字母要按照顺序排。。。lexicographer order

这轮最迷，印度大哥，在微软工作了八年了，刚加入现在这个组三个月。问了三十分钟 BQ 问题，又问了十五分钟 AWS，比如你自己理解的 lambda 后端怎么实现的？我以为这轮纯聊天，还剩十五分钟，来道 coding：input: "batman", K=3. Output: "aan"; input: "defab", K=3. Output: "dab"; 题目大意：给个 String 和 K，让你找长度为 K 的 subsequence of string 并且这个 subsequence 必须按照字母顺序排列。

看到例子，一脸懵逼，先问：第二个例子为什么 dab，这就不是排序了啊？面试官说：因为不管 ab，还是 ef，都小于 3，所以开头的字符也要。我 TM，二脸懵逼。继续问，等问清楚后，我说下我的思路，最后证明是错的，面试官提示：用 greedy 方法，问我知不知道 greedy 么？我说知道，但是不熟悉，从来没用过。面试官估计看出来我不会，开始带

着我走样例，他说完，我一半懂，一半没懂。这里如果有大神，欢迎讨论。最后超时了，问我有没有多余十五分钟，把 code 写下，我说有，可以的。就写 code，十五分钟也没写完，面试官说，时间有限，就这样吧。没问其他。

这轮面完，中间休息一个半小时，正常听说是，他们会讨论下前三轮表现，然后决定进不进行第四轮。感觉第三轮 code 没做出来啊。估计过不了吧。不过第四轮到时间，也没收到邮件，继续面。

第四轮：

印度人，group manager，说第二轮那个 HM 是给他 report。我面的职位也是他底下的职位。前二十分钟，给我讲他们组干啥的。听了半天，就是云的 authority。跟 AWS IAM 很像。然后问我希望是啥，我当然顺着说，我想去能接触 online service，high distributed system 的组。面试官频频点头。然后出了个题目：

DNSClient Cache:

1. 有个 API 可以叫 `getipaddress(String hostname)`
2. thread safe
3. have TTL，design correct evict
4. 有个 API 可以叫 `HostDetails，getHostDetail (String hostname)`

让设计这个 cache。开始就问各种策略，然后 assume 不是 thread safe 的，并且不管 TTL，只是 capacity 满了要怎么做。简化后就是蠡口 - 易思柳，大家都懂的。常规写出 get，put 方法。然后让去实现 DNSClient 这个 get 方法，注意这个方法里面，他要求，如果 local cache 就是你自己写的那个 cache 类里面，没有找到目标，请用 4 那个 hostdetail API 去 system call 拿到 host 的细节。再存入 cache 里面，就是让你 call put 方法。整个过程，也是面试官慢慢引导，交流才知道要这么做。刚开始并不知道这个干啥用。。。。

写完讨论了下 thread safe，这个部分只是答了下我对 lock 的理解，没让写 lock，然后问我如果让你在建的这么多个 class 里面加锁，哪个地方一定是不能加锁的。完全不知道，我说我知道双向链表可以加锁，但是这样效率低，因为多个 get method call 的话，整个链表加锁，会导致不同 key 的 get 都要互相等。面试官说：恩，这是个问题，不过你想想，哪里加上锁会有 risk？我不确定，我就试探性的说了：如果有 external service call 的地方不行？面试官说：对，在这个里面，就是 system call 那个 getHostDetails 地方不能上

锁。。。。我理解了他的意思，就说：恩，同样的道理，有 call 上锁会 risk 很大，其实我根本说不出来具体有啥 risk。大概就是理解 IO 上面用锁，肯定影响 performance。

这轮也是最后拖延了半个小时，但是总体看出来那个 group manager 还是满意的。然后问了个基本问题，就结束了。

周三面完，周四就告诉我过了，今天周五打电话谈 offer。最后 finalize 了 offer 的话会报到地里来的。

总结一下各种网上说的最近微软常考题目（蠢口）

5

15

23

34

38

46

49

124

127

218

240

259

305

393

545

773

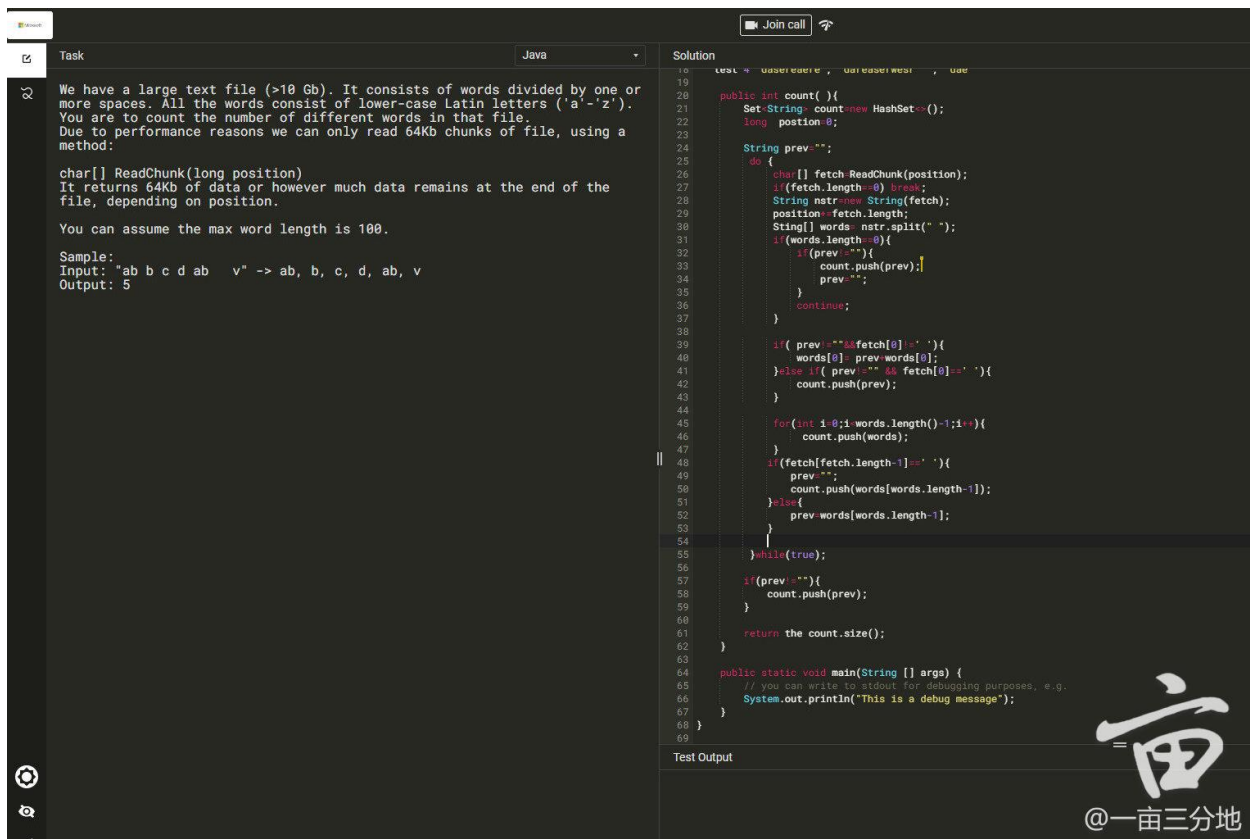
828

919

1206

1239

虽然不多 但在在于精。



The screenshot shows a coding interview interface. On the left, the 'Task' panel contains the following text:

We have a large text file (>10 Gb). It consists of words divided by one or more spaces. All the words consist of lower-case Latin letters ('a'-'z'). You are to count the number of different words in that file. Due to performance reasons we can only read 64Kb chunks of file, using a method:

```
char[] ReadChunk(long position)
// It returns 64Kb of data or however much data remains at the end of the file, depending on position.

You can assume the max word length is 100.
```

Sample:
Input: "ab b c d ab v" -> ab, b, c, d, ab, v
Output: 5

On the right, the 'Solution' panel shows a Java code snippet for counting unique words:

```
public int count() {
    Set<String> count = new HashSet<>();
    long position = 0;
    String prev = "";
    do {
        char[] fetch = ReadChunk(position);
        if (fetch.length == 0) break;
        String natr = new String(fetch);
        position = fetch.length;
        String[] words = natr.split(" ");
        if (words.length == 0) {
            if (prev != "") {
                count.push(prev);
                prev = "";
            }
            continue;
        }
        if (prev != "" && fetch[0] != ' '){
            words[0] = prev + words[0];
        } else if (prev != "" && fetch[0] == ' '){
            count.push(prev);
        }
        for (int i = 0; i < words.length - 1; i++) {
            count.push(words[i]);
        }
        if (fetch[fetch.length - 1] != ' '){
            prev = "";
            count.push(words[words.length - 1]);
        } else {
            prev = words[words.length - 1];
        }
    } while (true);
    if (prev != "") {
        count.push(prev);
    }
    return count.size();
}

public static void main(String[] args) {
    // you can write to stdout for debugging purposes, e.g.
    System.out.println("This is a debug message");
}
```

At the bottom right, there is a watermark logo and the text '@一亩三分地'.

1, Describe one situation where you make use of a new technology to work on a project.

2, Describe how you can learn a new technique fast.

3, Describe a situation where you succeed by collaborating with other people, or a bad collaboration leading to a bad outcome.

4, What Microsoft product do you think you can apply your skills on?

[hide=120]

5, Given a string, implement a function that replaces all blank spaces by “%20” with least possible memory space and time.

6, In the “Candy crush” game, given a 2D array that stores the type of the candy at each position of the table. Write a function that tells which candies are going to explode. If the same type of candy forms a straight line with size larger than 2, all candies on that line should explode. Follow up: what to do when the size of the result is huge?

7, Given a string that stores an arithmetic expression, for example $2+4*2-3/8$ assuming that the numbers all have only 1 digit. Write a function to compute the calculation result of the expression. What is the time complexity? How can you improve?

8. Given a binary tree, design a method to print it out upside down.
9. Implement a finite-size queue class with an array. (Follow up: test the push and pop function on a small size queue. How do you deal with pop() when the queue is empty? How to deal with corner cases so that your library is stable under all inputs).
10. Implement a multiplication method for two integers. You must not use the built-in multiplication and division operators.
11. (Brain teaser) Lily leaves work everyday at 5:30pm, after going to the supermarket, she catches the 6pm train which arrives at the train station at 6:30. Her husband then picks her up at 6:30 and drives home. One day, Lily left work at 5:25pm and caught the 5:30 train which arrives at station at 6:00. She then walks home but her husband still follows the original schedule and drove out to pick her up. Lily met her husband on her way home and arrives 10 minutes earlier than normal. Based on the above information, calculate the time when Lily is picked up by her husband.

店面：

1. 利口腰遛舞。需要考虑到一些边缘情形
2. HM 面的 一道题 义乌散
3. path sum: 和[刷题](#)网题目差不多, 返回最大值即可, 不一定要 root to leaf.
4. 物流
- 5.

Microsoft给的是

🔒 以下内容需要积分高于 50 您已经可以浏览

Base: 110k
RSU: 70k
sign on: 15k

还没有compete 过，但base应该不变，最多stock增加到120k左右？

Doordash:

🔒 以下内容需要积分高于 50 您已经可以浏览

Base: 135k
RSU: 175k / 4
sign on: 20k

base salary 一年工资:	110500
年终奖金Yearly Perf Bonus:	1dk
Equity(RSU/Option)总价值(通常给四年的):	120k
Equity种类:	RSU
Equity Vesting schedule:	25/25/25/25
签字费sign on bonus第一年:	点击查看
签字费sign on bonus第二年:	17500
搬家费 relocation:	~8000
是否满意，打算去嘛:	不确定
应届or在职跳槽:	fresh grad应届毕业生
目前所在地区:	美国其他地区
目前最高学历:	本科
相关工作经验范围:	短暂实习或全职不超过3个月

🔒 以下内容需要积分高于 5 您已经可以浏览

今年59在seattle都是这个base，别的地区根据tax会有小difference。stock最高好像有150k但好像是60不是59，signon最高是50k，最低看到20k的也有。总的来说还可以，还没有negotiate。听说软家以后工资涨幅不行，不过工作是真的轻松，组也很好。goog和fb好像都freeze ng了，虽然现在还没接，但估计最后就去了。fun fact我去seattle面试的时候接我的uber司机说我一看不是Amazon的，因为我看上去没有depressed。