

# 总体

- 态度：不卑不亢，有自信
- 发音：慢而清晰
- 书写：行距适中，头要对其，字要始终，工整，缩进4个空格，行距适中

## BQ问题

- 背熟练，眼睛要对上，自然
- 简明回答，诱导对方进一步提问

## 确认题目要求

- 具体步骤：
  - 复述确认sample input/output
  - data structure
    - 链表有几个环等
  - corner case: (corner case包含edge case)
    - 输入：内容状态 ('空1重序') + 取值范围(正负0, 整数)
      - LinkedList: 改变node数值, 有序无序
      - 二叉树: 值是什么类型的
    - 输出: '空1重序' 输入情况的结果/条件不满足, 输出原输入数/结果数字重复怎么输出
  - 明确: 注重time complexity or space complexity?
- 面试官提问环节:
  - 先把自己会的说出来, 再求教于面试官补充
  - 与面试环交流有歧义, 要及时纠正, 不要就坡下驴

## 写出数据结构和算法, 解释

- 整体步骤:
  - 暴力+优化的层次分析
    - linklist: list法 + two pointer法等
- 具体步骤:
  - 提出可以尝试某一种idea (brute force or others), 说要思考, 并解释简要原因
  - 写出idea及method(如有必要, 请画图)
    - idea: 方法名 brute force / two pointer / change next pointer
    - method:
      - 解释的时候用例子更易懂
      - 包含定义好的变量名的, traversal/recursion等过程名的过程表述。e.g. traversal the linklist to save node val in node\_list
      - 用case1 case2 case3表示各个条件情况
      - 要简单一些。eg. 只要说traverse/scan index即可, 不用说from i to j,
  - 给出space/time complexity
    - 对于dic等space comp, 要具体分析。由于题目条件, 可能存在只有有限个(2个等), 故此时sO(2) = sO(1), 而不是通常的O(N)
- tips:
  - 注意编号的清晰
  - 为了保险, 用for i in range(len(x)), 而不是 for elem in list
  - 如果发现edge case不完全, 要写到2部分, 不要在本部分插入
  - 套模板时, 要在此处想清楚, 不要等到写代码时再想

## code之前:

- 'should I implement it in code? or you want me to continue to optimize it?'
- 'we have ... minutes. It seems that I can finish it in time.'

## right coding

- 具体步骤:
  - 强调需要使用**class**，并对**class**和**function**命名，加功能注释及特殊输入格式的提醒
  - 写出框架，再擦掉即可：

```
def func(self,input): #inp: XXX, outp: numb of string...
    res = 0
    方框
    return res
```
  - 先写主函数，再写**help**函数
  - 变量名定义要有意义，不要后补: temp/ans/res
  - 书写代码，并同时书写注释:
    - 对复杂操作和无法一目了然的代码，再行尾添加描述代码目的的注释
    - 不要描述代码，比如traversal i from 1 to 19
    - 针对'method'中列出的case，针对性的注释case1/case2/case3即可
- 状态:
  - 写代码时不要自言自语，想什么就跟考官交流，说我要做什么什么了
- tips:
  - 发现method不完全，需要添加内容时：同时添加method和code
  - 遇到特别大的数进行运算时，可以说“当使用python时，可以避免像c++那样的数据溢出问题（overflow）”
  - 求list中的长度，j-i即可
  - 保证书写出代码的有效性，e.g.a.next.next,要保证a.next的有效，故在判断条件中要书写出来

## 跑test case

- 具体步骤:
  1. 跑corner case
    - 跑面试官的case
      - 先验主函数，再验证helper function
      - test要清晰的隔绝开，不乱
    - 再跑自己的case
- 格式: 前后风格格式要统一，要易懂
- 题目类型test:
  - 链表: 画图指示箭头 + table
  - array: 分为两个部分，总结果表示各个关键变量的值，用table表示；小结果写在code的右边，即写即擦，展现具体的计算过程

## 发音和书写

- 发音: 慢而清晰
- 书写: 行距适中，头要对其，字不要太小，缩进4个空格，行距可以小!!!

# 考官交流

- 问了必答，比如在idea区域问space comp时，也必须要回答，虽然不清楚具体的结果，也要说我预计xxx，下面可能会有改变
- 不会做时，考官会给与提示，此时要做到：
  - 及时跟进自己的想法，明白他说的细节
  - 他说完后，要整理明白这段所有的思路，并叙述出来，与他交流。
- following up没时间写时，可以只讲思路，怎么做就行。等面试结束后，可以查面试官邮箱，把具体方法发到他邮箱里。

# 备考要求

- 熟练掌握每种题型对应的brute force和常用方法
- 针对leetcode，想清楚某系列题目之间的相关关系，I和II之间区别与联系
- 某些特殊方法的数学证明要熟悉