

# Practice

## 任务一：

### 1. 两数之和

- 分析题目：将 `nums` 中和等于 `target` 的两个数字筛选出来
- 选择方法：
  - i. 用 `for` 循环将 `nums` 遍历两次
  - ii. 判断和是否等于 `target`
- 检查逻辑漏洞：如果在第二次循环时不剔除第一次循环过的数，会导致重复
- 参考答案分析：可用哈希表（`enumerate` 函数）来减小时间复杂度

### 9. 回文数

- 分析题目：判断将数字倒序读与顺序读是否相等
- 第一次选择方法：
  - i. 将数字拆分成列表
  - ii. 进行顺序和倒序判断
- 检查逻辑漏洞：若数字为负数则无法判断
- 第二次选择方法：
  - i. 将数字转化成字符串
  - ii. 利用序列倒序的方法将字符串反转
  - iii. 判断
- 参考答案分析：将数字对半翻转，并判断是否与前半段数字相同

### 1773. 统计匹配检索规则的物品数量

- 分析题目：检查 `items` 列表中每个元素列表中的 `ruleKey` 是否等于 `ruleValue`，并返回元素列表的数量
- 选择方法：
  - i. 创建一个字典来表示 `ruleKey` 在每个元素列表中的索引
  - ii. 设置一个 `num` 来记录符合条件的元素
  - iii. 用 `for` 循环遍历 `items`
  - iv. 判断
- 参考答案分析：用哈希表将 `ruleKey` 的索引表达出来，用推导式来遍历 `items`，最后 `sum` 求和

### 2114. 句子中最多的单词数

- 分析题目：找出句子中单词数最多的句子，返回其单词数
- 选择方法：
  - i. 设置一个 `lst` 统计每个句子的空格数
  - ii. `for` 循环遍历列表，并计算空格的数量
  - iii. 用 `max` 选出最大空格数，再加一得到单词数，并返回

### 1252. 奇数值单元格的数目

- 分析题目：将一个 `m*n` 且初始参数为 0 的矩阵按照一定格式在行和列上分别加一，并返回矩阵中奇数的数量
- 选择方法：
  - i. 先建立一个初始矩阵
  - ii. 用 `for` 循环先遍历 `indices` 列表以获得需要处理的行和列
  - iii. 用 `for` 循环遍历矩阵，并根据上步得到的数据进行加一处理
  - iv. 用 `for` 循环遍历得到的新矩阵，并记录奇数的数量

- 错误总结: i. 不能用 `[[0 * m] * n]` 来获得矩阵, 否则会形成将 `0 * m` 的结果重复 3 遍的列表  
ii. 不能用 `[[0] * m] * n` 来获得列表, 会形成浅拷贝, 使形成的列表有关联
- 参考答案分析: i. 建立横行 `rows` 和竖行 `cols` 两个列表  
ii. 根据 `indices` 列表在将上一步的列表进行处理  
iii. 将处理后的 `rows` 和 `cols` 中的数字分别相加并判断是否为奇数

## 1652. 拆炸弹

- 分析题目: 根据 `k` 的值来将 `code` 中的数字依次进行处理, 并得到密码
- 选择方法: i. 建立一个列表记录密码  
ii. 判断 `k` 的值  
iii. 利用切片按题目规律来提取数据并求和
- 参考答案分析: i. 使用滑动窗口的方法 (将 `code` 自身相加)  
ii. 设置好遍历的范围, 并根据 `k` 的值来处理数据

## 977. 有序数组的平方

- 分析题目: 将给定数组中的数字全部平方, 并将其降序排列
- 选择方法: i. 用列表推导式将给定数组平方化  
ii. 用 `sort` 将数组排序

## 1184. 公交站间的距离

- 题目分析: 求出给出两个公交站间的最小距离
- 思路转换: 将题目转换成直接算出两站距离, 再用总距离减去
- 选择方法: i. 判断 `start` 和 `destination` 的大小  
ii. 按将 `start` 到 `destination` 切片 `distance`  
iii. 将切片求和  
iv. 用总距离减去切片和, 并判断出最小量

## 1431. 糖果最多的孩子

- 题目分析: 判断列表每个数字加上某个数字后是否为列表中最大
- 选择方法: i. 建立一个表格用于记录判断结果  
ii. 用 `enumerate` 函数获得各数字的索引  
iii. 利用索引获得相应的值并加上所给值, 并与列表中的最大值相比

## 605. 种花问题

- 题目分析: 在非相邻的花盆中种花, 并判断是否可以种入 `n` 盆花
- 选择方法: i. 在“花坛”两头加上 0 建立“新花坛”  
ii. 从左往右遍历“花坛”, 在符合条件的地方将“花”插下  
iii. 记录下所插“花”的数量, 并与 `n` 相比
- 参考答案分析: i. 从贪心角度, 尽可能种更多的花  
ii. 利用两花间空余位置 (`p`) 的奇偶来计算可种植的花  
iii. 奇:  $(p - 1) / 2$  偶:  $(p - 1) // 2$   
iv. 用循环遍历花坛, 并记录种植花的数量

## 面试题 16.11. 跳水板

- 题目分析：将 shorter 和 longer 两种板取出共 k 块，并输出所有可能出现的长度
- 选择方法：i. 建立空列表用于记录排列结果
  - ii. 将 shorter 板从 k 块开始减少至 0，longer 板从 0 开始增加至 k 块，并将总长度添加至列表
- 错误分析：未考虑 k=0 和 shorter=longer 两种情况

#### 409. 最长回文串

- 题目分析：将字符串 s 可以构成的最长回文串（即左右对称的字符串）的长度输出
- 方法选择：i. 建立一个列表 lst1，将字符串 s 中的字母逐个提出
  - ii. 用 for 循环和 lst1 集合化去重，将 lst1 中的字母数量全部数出并记录在 lst2 中
  - iii. 将 lst2 中的奇数全部减一后，将 lst2 求和并返回结果
- 错误分析：回文串中间可以有单个字母

## 任务二：

#### 12. 整数转罗马数字

- 题目分析：输入一个整数，将其转化为罗马数字，其中一般罗马小的数字在大的数字右边，除了 4 和 9 等特殊数字，因此可将整数分为 (0, 3], (3, 5], (5, 8], 9 四个区间进行判断
- 方法选择：i. 建立列表，分别记录罗马数字中的个位，十位…
  - ii. 将整数的每个数字取出，并倒序（显然从个位开始更加方便），得到 lst 列表
  - iii. 建立空字符串，用于记录所得到的罗马数字
  - iv. 遍历 lst，判断其所在区间，并做出相应处理
  - v. 将得到的罗马数字倒序并输出其长度
- 参考答案分析：i. 建立列表，记录每个特殊的值和对应罗马符号（1, 4, 5, 9）
  - ii. 将 num 从大到小开始减去特殊值，并添加相应的罗马符号
  - iii. 直到 num 减为 0 时，输出得到的罗马数字

#### 31. 下一个排列

- 题目分析：给出一个列表 nums，给出比 nums 字典序大的下一个列表，若已经是最大字典序，则将其改为最小字典序状态，且 nums 必须原地修改，只能使用额外的常数空间
- 第一次方法选择：从后往前遍历 nums，如果后一位数比前一位数大就交换位置
- 错误分析：数字改变过多，不为‘下一排列’
- 第二次方法选择：i. 第一步先判断是否为最大字典序，是则将 nums 倒序并输出 None，否则进入第二步
  - ii. 第二步从后往前（除了第一位数）遍历 nums，若后数比前数大，则直接输出 None，否则进入第三步
  - iii. 第三步取出第一个比第一位数大的数并交换位置，在将剩下的数字顺序排列后，将取出的数字添加到第一位
- 错误分析：只考虑了第一位的改变，若 nums 超过 3 位则不符合条件
- 第三次方法选择：i. 第一步仍为判断是否为最大字典序与第二种方法相同
  - ii. 用两个循环来将 nums 从后往前切片，并判断顺序排列后的切片中是否有比前一位数大的数，若有则将第一个比前数大的数与前数交换位置

## 540. 有序数组中的单一元素

- 题目分析：
  - i. 在数组 `nums` 中每个元素出现两次，只有一个单一元素，要求找出这个元素，且时间复杂度满足  $O(\log n)$ ，空间复杂度满足  $O(1)$ 。
  - ii. 由时间复杂度可知题目要求用二分查找
- 寻找规律：发现单一元素所在位置与最中间的数字 (`nums[x]`) 和数字数量的奇偶性有一定关系，可分为四种情况：
  - i. 与右数不等，右边数字数量为偶数
  - ii. 不等，奇数
  - iii. 相等，偶数
  - iv. 相等，奇数
- 方法选择：因此可根据情况的不同来删除相应的数组切片，但保持留下的数组始终由出现两次的元素和单一元素组成
  - i. 用 `while` 循环使数组长度等于 1 时停止
  - ii. 对 `nums` 进行删除处理
  - iii. 输出 `nums[0]`

## 任务三：

### 37. 解数独

- 题目分析：在输入的数组棋盘中填入数字  
数独规则：
  - i. 同一行和同一列上不能出现重复数字
  - ii. 九宫格内不能有重复数字
  - iii. 数字只能是 1-9
- 选择方法：显然需要枚举出多种可能的答案，并输出唯一答案，此时使用回溯法
  - i. 定义一个函数用于检查三种情况是否有重复数字，并返回布尔值
  - ii. 定义一个主函数用于遍历数字（回溯）
- 函数构建：
  - i. 检查函数：
    - a. 行和列直接通过所在坐标来进行遍历检查
    - b. 九宫格用  $(x // 3) * 3$  和  $(y // 3) * 3$  来得到所在九宫格坐标后遍历检查
  - ii. 主函数：
    - a. 首先对 `x`, `y` 进行判断，看是否需要换行，并进行处理
    - b. 判断在  $(x, y)$  位置是否已有数字，若有则跳过该位置
    - c. 在上述处理后遍历 1-9 数字并检查是否符合填写要求
    - d. 再次调用主函数，并使 `x` 加一，进行第二步判断
    - e. 若在之后有不符合条件的数字则返回该步进行初始化处理