

剑指 offer 11 . 002 . 二进制加法

从串尾开始一位一位相加，注意进位

时: $O(n)$

空: $O(n)$

539. 最小时间差

- ① 给时间列表按小到大排序
- ② 若前后有相同时间则返回 0
- ③ 计算所有相邻的时间差，记下最小的
- ④ 首尾两时间也需要计算。

时: $O(n \log n + n)$ 空: $O(1)$

剑指 offer 11. 018. 有效的回文

首尾双指针，比较首尾每个字符的lower()是否相同

时: $O(n)$

空: $O(1)$

219. 存在重复元素 II

哈希表记录每个元素的下标，若某个元素已在哈希表内，则
算 $\text{abs}(i-j)$ 是否 $\leq k$ ，是则返回，否则更新该元素
在哈希表的下标。

时: $O(n)$

空: $O(1)$

剑指 offer 11. 068. 查找插入位置

二分查找

while $\text{tail} > \text{head}$:

$\text{mid} = (\text{head} + \text{tail}) // 2$

若 $\text{nums}[\text{mid}] = \text{target}$: 返回

若 $\text{nums}[\text{mid}] > \text{target}$: $\text{tail} = \text{mid} - 1$

若 $\text{nums}[\text{mid}] < \text{target}$: $\text{head} = \text{mid} + 1$

时: $O(\log n)$

空: $O(1)$

剑指 offer 11. 078. 合并排序链表

用小根堆将每个链表的链首按 val 值，链表序号存起来，

pop 出小根堆顶，再根据序号 push 进新的值组合，将 pop 出的值拉成链

时： $O(n \log n)$

空： $O(n)$

2029. 石子游戏 IX

- ① 每个值相当于 0, 1, 2
- ② 统计 0, 1, 2 的个数
- ③ 若 0 的个数为偶数，相当于没有 0。
- ④ 若 0 的个数为奇数，相当于只有 1 个 0。
- ⑤ 没有 0 的情况，只要 1 和 2 都至少有 1 个，A 赢
 有一个 0 的情况，1 和 2 的个数差要大于 2, A V

时: $O(n)$

空: $O(1)$

1345. 跳跃游戏 IV

- ① 哈希表记录值对应的下标
- ② visited 记录下标是否遍历。
- ③ q 是访问队列 (下标, 步数)
- ④ 遍历过后要过河拆桥

时: $O(n+n+n)$

空: $O(n)$

剑指 offer 11. 101. 分割等和子集

① 若 数组和为奇数 \rightarrow False

② 动态规划：

for target 到 $y-1$:

$$x = t - y$$

若 $dp[x] = \text{True}$, $dp[t] = \text{True}$

时: $O(n^2)$

空: $O(n)$

1332. 删除回文子序列

① 若字符串本身是回文串，返回 1

② 若不是回文串，只需 2 次删除即可

时: $O(n)$

空: $O(1)$

剑指 offer 11. 033. 变位词组

- ① 将每个词都用排序方法组出变位词 用 哈希表 记录下来
- ② 再遍历将哈希表中分好类的词取出

时: $O(nm + n)$

空: $O(nm)$

2034. 股票价格波动

- ① 哈希表 记录股票时间戳与对应的价格 ; 时: $O(n)$
- ② 有序表 用二分插入与二分查找 整理 ; 时: $O(n \log n)$
- ③ 最新股票价格 用二元式记录 ; 时: $O(1)$

空: $O(n)$

剑指 offer 11. 098 . 路径的数目

动态规划：第一行、第一列为 1，其余 $dp[i][j] = dp[i-1][j] + dp[i][j-1]$

时: $O(n^2)$

空: $O(n^2)$

剑指 offer 11. 049. 从根节点到叶节点的路径数字之和

dfs，到叶节点的时候将数字累加起来

时: $O(n)$

空: $O(\log n)$

1688 · 比赛中的配对次数

返回上一

时: $O(1)$

空: $O(1)$

剑指 offer 11. 019. 最多删除一个字符得到回文

首尾双指针，两指针所指字符若相同，双向奔赴；

若两指针所指字符不同，则 $s[\text{head}+1: \text{tail}]$ 与 $s[\text{head}: \text{tail}]$ 再所一遍上述操作，二者都存在不同，则 False，否则 True.

时: $O(n)$

空: $O(n)$

剑指 offer 11. 072 . 求平方根 .

从 0 开始，逐一尝试，直到平方大于等于 x .

时: $O(\log n)$ 空: $O(1)$

1996. 游戏中的弱角色

① 先将防御正序排序，再将攻击力倒序排序

② $\max = [\text{prop}[0][0], \text{prop}[0][1]]$

若 $\text{item}[1] > \max[1]$: 若 $\text{item}[0] = \max[0]$, R) $\text{item}[1] = \max[1]$

否则 $\max[0]$ 也改为 $\text{item}[0]$

若 $\text{item}[1] < \max[1]$: 若 $\text{item}[0] < \max[0]$ 则 $\text{res} + 1$

时: $O(n \log n + n)$

空: $O(n \log n)$

1765. 地图中的最高点

多源广度优先遍历：先将水岸位置标出，再以水岸扩展开来

时： $O(mn)$

空： $O(mn)$

884. 两句话中的不常见单词.

2个哈希表分别统计两句话中单词出现次数，其中出现一次的单词再判断是否在另一句话中出现

时： $O(n)$

空： $O(n)$

1342. 将数字变成0的操作次数.

奇数 -1 , 偶数 /2 , 直至数为0 .

时: $O(n)$

空: $O(1)$

剑指 offer 11. 03. 直方图最大矩形面积

单调栈：从左到右遍历，将比当前高的都取出来

```
1 class Solution:
2     def largestRectangleArea(self, heights: List[int]) -> int:
3         heights.append(0)
4         n = len(heights)
5         stk = []
6         res = 0
7
8         for i in range(n):
9             x = heights[i]
10            while stk and heights[stk[-1]] >= x:
11                h = heights[stk.pop(-1)]
12                l = stk[-1] + 1 if stk else 0
13                r = i - 1
14                cur = (r - l + 1) * h
15                res = max(cur, res)
16            stk.append(i)
17        return res
```

时: $O(n)$

空: $O(n)$

1763. 最长的最美子字符串.

遍历以每个字符，看是否为最美字符串以及比较长度.

时: $O(n^2)$ 空: $O(n)$

剑指 offer 11. 040. 矩阵中最大的矩形

- ①先用 dp 算出每一行的高度
- ②再用单调栈算出直方图最大面积

时: $O(mn)$

空: $O(mn)$

1219. 黄金矿工

dfs : 注意每次回溯以及开启深度优先遍历之后要还原 visited 数组

时: $O((mn)^2)$

空: $O(mn)$

剑指 offer 11. 001. 整数除法

```
● num << 1 = num * 2;  
● 1 << n = 2 ^ n  
● 计算 64 / 3 = ?  
● 3 << 0 = 3 = (1 << 0) * 3;  
● 3 << 1 = 6 = (1 << 1) * 3;  
● 3 << 2 = 12 = (1 << 2) * 3;  
● 3 << 3 = 24 = (1 << 3) * 3;  
● 3 << 4 = 48 = (1 << 4) * 3;  
● 3 << 5 = 96 > 64  
● 得到 maxShift = 4  
➤ int res = 0;  
➤ 64 - (3 << 4) = 64 - 48 = 16  
res += (1 << 4) = 16  
➤ 16 - (3 << 2) = 16 - 12 = 4  
res += (1 << 2) = 16 + 4 = 20  
➤ 4 - (3 << 0) = 4 - 3 = 1  
res += (1 << 0) = 20 + 1 = 21  
➤ 64 / 3 = 21
```

时: $\mathcal{O}(1)$

空: $\mathcal{O}(1)$

1748. 唯一元素的和.

哈希表将数组中元素出现次数统计起来，对于第一次出现的数加起来，若第二次出现则需减去，其余则不管。

时: $O(n)$

空: $O(n)$

剑指 offer 11. 091. 粉刷房子

dp :

$$dp[i+1][0] = costs[i][0] + \min(dp[i][1], dp[i][2])$$

时: $O(n)$

空: $O(n^2)$

1405. 最长快乐字符串.

贪心法：当 res 长度 < 2 或 $\text{res}[i-1] \neq \text{res}[i-2]$ 时，随便加；

当 $\text{res}[i-1] = \text{res}[i-2]$ 时，不能加与 $\text{res}[i-1]$ 相同的字符。

时： $O(n)$

空： $O(1)$

1984. 学生分数的最小差值

① 排序

② 计算 $\text{nums}[i] - \text{nums}[i-k]$ ，保留最小差值

时: $O(n \log n + n)$ 定: $O(n)$

540. 有序数组的单一元素

- ① 二分法看中间元素是否是单一元素
- ② 若否，则看该组元素的开头下标是否为偶数，是则说明单一元素在后，否则单一元素在前

时: $\mathcal{O}(\log n)$

空: $\mathcal{O}(1)$

1380. 矩阵中的幸运数.

找到每行中最小数 同时看其是否是该列最大数

时: $O(n^2)$

空: $O(n)$

金) 指 offer 11. 086. 分割回文子字符串.

用 回溯法 分割 所有可行的子字符串.

时: $O(n^2)$ 空: $O(n)$

688. 骑士在棋盘的概率

① $dp[n][n][k+1]$, 先在 $dp[i][j][0]$ 都赋 1 ($i, j = 0 \dots n-1$)

② 随后通过 $dp[nr][nc][i] += dp[r][c][i-1] / 8$ 更新

③ 最后返回 $dp[row][column][k]$

时: $O(k \times n^2)$

空: $O(k \times n^2)$

1791. 找出星型图的中心节点。

只有中心节点会与其他节点相接，其余节点的度为 1

返回前一个对中重复出现的节点即可

时: $O(1)$

空: $O(1)$

769. 煎饼排序

- ① 先找到待排的最大元素，下标为 index
- ② 逆置 $\text{arr}[0: \text{index}+1]$ ，再逆置 $\text{arr}[0: n]$ ，最大元素就位

时: $O(n^2)$

空: $O(1)$

71]. 1比特和2比特字符串

$i = 0, 1, \dots, n - 1$

若 $\text{bits}[i] = 1$, $i + 2$

若 $\text{bits}[i] = 0$, $i + 1$

时: $\mathcal{O}(n)$

空: $\mathcal{O}(1)$

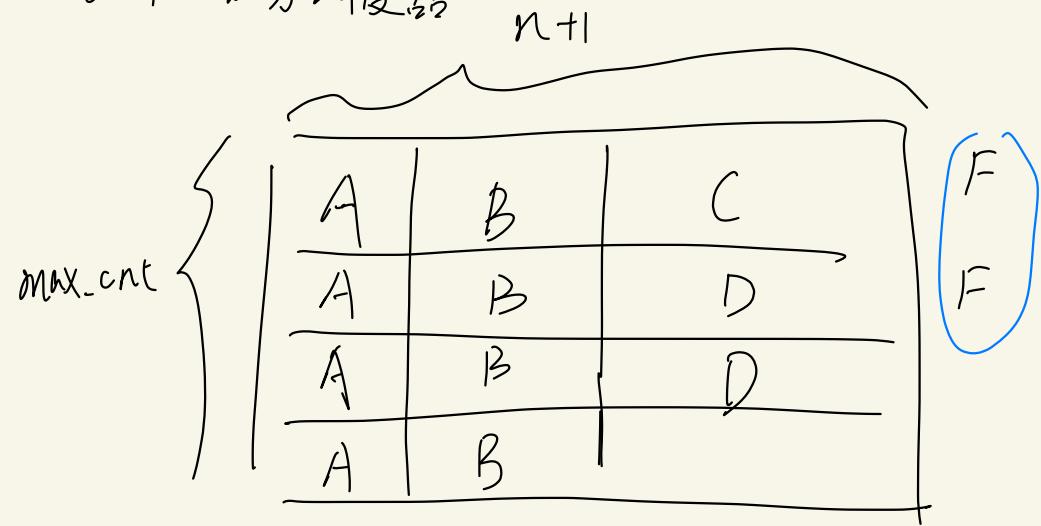
838.推多米诺

- 双指针思路：
- ① 若右指针遇上 ‘.’，右指针右移
 - ② 若左指针‘L’，右指针‘R’，则中间的骨牌都不会受影响
 - ③ 若左 右指针都是一样的，则中间骨牌都倒向一个方向
 - ④ 若左指针‘R’，右指针‘L’，则从两边向中间倒

时： $O(n)$

空： $O(n)$

621. 任务调度器



时: $O(n)$

空: $O(n)$

917. 仅仅反转字母

① 遍历第一遍把字母记录下来

② 遍历第二遍遇上字母则将记录的字母倒序放置，其他则保留。

时： $O(2n)$

空： $O(n)$

394. 字符串解码

multi, res, stack

遍历 如遇 '[' , stack.append([multi, res]), multi, res 重置

如遇 ']' , stack.pop() , res = last_res + multi * res

如遇 数字 , multi = multi * 10 + 数

遇字母 , res += ch

时: $O(n)$

空: $O(n)$

1706. 球会落何处

- ① 当 $\text{index} = 0$, $\text{grid}[i][\text{index}] = -1 \Rightarrow \times$
- ② 当 $\text{index} = n-1$, $\text{grid}[i][\text{index}] = 1 \Rightarrow \times$
- ③ 当 $\text{grid}[i][\text{index}] = 1$, $\text{grid}[i][\text{index}+1] = -1 \Rightarrow \times$
- ④ $\nexists \text{ grid}[i][\text{index}] = -1$, $\text{grid}[i][\text{index}-1] = 1 \Rightarrow \times$

时: $O(n^2)$

空: $O(n)$

537. 复数乘法

① 将实部与虚部提出来

② 注意 $i \times i = -1$

时: $O_{(1)}$

空: $O_{(1)}$

416. 分割等和子集

① 算 total , 若 total 奇 \rightarrow False

② target = total / 2

dp = [True] + [False] \times target , 子集能凑出的和

时: $O(m \cdot n)$

空: $O(n)$

2016. 增量元素之间的最大差值

遍历一次，若 $\text{nums}[i] < \text{mini}$ ，记录下新 mini 以及 index ，

否则 $\text{res} = \max(\text{res}, \text{nums}[i] - \text{mini})$

时: $O(n)$

空: $O(1)$

494. 目标和

$$dp ; dp[i] += dp[i + 2 \times num]$$

时: $O(n^2)$

空: $O(n)$

1601. 最多 可达成的换楼请求数目

回溯法：对所有请求都进行考虑，看其是否可行，可行则记下请求数目，返回最大的请求数目。

时： $O(n^2)$

空： $O(n)$

553. 最优除法

因为每个元素都大于 1，要使表达式结果最大， $x \div y$ 即只要使 y 尽可能小即可。所以除了首元素，后面元素连除即可

时: $O(n)$

空: $O(1)$

312. 截气球 (不明白)

dp : ① 数组头尾先插 |

② 对于开区间 (i, j) , 罢找其中 k 个气球所得的币为

$$dp[i][k] + val[i] \times val[k] \times val[j] + dp[k+1][j]$$

时: $O(n^3)$

空: $O(n^2)$

6. Z 字形变换

① 若 $\text{numRows} = 1$ ，则不需要进行任何操作

② 设 numRows 个字符串，来回添加字符

时: $O(n)$

空: $O(n)$

42. 接雨水

dp ; 统计每一格左右最高值中的较小者与自身相减的高度

时: $O(2n)$

空: $O(2n)$

564. 寻找最近的回文数

特殊情况：① $n < 10$ 或 $n = 10^a$ $\Rightarrow \text{return } n-1$

② $n = 11 \Rightarrow \text{return } 1$

③ $n = 9\dots991 \Rightarrow \text{return } n+2$

正常： n 前一半替换 n 的后一半，排除自身。

时: $O(n)$

空: $O(n)$

258. 各位相加

若 $\text{num} = 0$, return 0

若 num 为 9 的倍数, return 9

若 num 不为 9 的倍数, return num \% 9 .

时: $O(1)$

空: $O(1)$

221. 最大正方形

dp : 转移方程 : $dp[i][j] = 1 + \min(dp[i-1][j-1], dp[i-1][j], dp[i][j+1])$

时: $O(n)$

空: $O(n)$

2104. 子数组范围和 (若 $\text{nums}[i] = \text{nums}[j]$ ，若 $i < j$ ，则 $\text{nums}[i] < \text{nums}[j]$)

① 每个 $\text{nums}[i]$ 成为区间最大值次数 k_1 ，以及最小值次数 k_2 ， $(k_1 - k_2) \times \text{nums}[i]$ 则为该数为总范围和的贡献。

② 单调栈统计一个数左右最近的比之大或小的数的下标

③ 若一个数 $\text{nums}[i]$ 左右比之大的数下标为 p, q

则 $(i - p) \times (q - i) \times \text{nums}[i]$ ，小的亦然。

④ $\text{res} = \text{sumMax} - \text{sumMin}$

时: $O(n)$

空: $O(n)$

166. 分数到小数

① 若 分子可整除分母，则 直接返回 $\text{str}(\text{分子} \div \text{分母})$

② 若 分子与分母不同号，结果要加负号

③ <1> 先算整数部分

<2> 再算小数部分

<3> 小数部分每次的余数用 哈希表 记录下来，用于找循环节

时： $O(n)$

空： $O(n)$

521. 最长特殊序列 I

若 $a = b$, 返回 -1;

若 $a \neq b$, 返回较大的长度.

时: $\mathcal{O}(n)$

空: $\mathcal{O}(1)$

2100. 适合打劫银行的日子

- ① dp 统计所有日子递增与持平的日子
- ② 对于 $time \in [length - time]$ 索引的日子看前后是否满足打劫要求

时: $O(n)$

空: $O(n)$

504. 七进制数

若 $|num| < 7$ ，直接返回 $\text{str}(num)$

否则 采用除7取余算得七进制数

时: $O(n)$ 空: $O(1)$

2055. 蜡烛之间的盘子

- ① 统计每个盘子的左右最近的蜡烛的下标
- ② 统计每个下标的蜡烛前缀和

$$时间: O(3n+m)$$

$$空间: O(3n)$$

798. 得分最高的最小轮调. (不懂)

差分数组 : $score[i]$ 表示 k 步后的得分.

关键点 : ① $score[(i + n - \text{mumss}[i] + 1) \% n] -= 1$

② $score[i] += score[i + 1] + 1$

时: $O(2n)$

空: $O(n)$

589. N叉树的前序遍历

递归法 or 迭代法

时: $O(n)$

空: $O_c(\log n)$

2049. 统计最高分的节点数目

dfs :

① 先统计节点的子节点

② dfs 统计节点的得分 , $\text{left} \times \text{right} \times (n - \text{left} - \text{right} - 1)$

时: $O(n)$

空: $O(n)$

599. 两个列表的最小索引总和.

① 哈希表映射 $list_1$ 中餐厅与其索引

② 遍历 $list_2$ 若遇上与 $list_1$ 重合的餐厅，则看索引和的大小，比 $minsum$ 小

则重置 res，若等于 $minsum$ 则加在 res 上

时: $O(2n)$

空: $O(n)$

2044. 统计按位或能得到最大值的子集数目

- ① 先将所有元素都按位或得到最大值
- ② dfs 计算所有可能的子集 按位或的值，统计其中等于最大值的数目

时: $O(n + n^2)$ 空: $O(n)$

432. 全 $O(1)$ 的数据结构

双向链表 + 哈希表

链表每个节点记录子字符串以及对应的次数

哈希表记录每个子字符串对应的节点

时: $O(1)$

空: $O(2n)$

720. 词典中最长的单词

- ① 先将词典排序
- ② 遍历词典，若词的前缀已出现在哈希表中，继续下一步
- ③ 看词的长度是否比所记录的最大长度长，是则继续
- ④ res 记录下该词同时更新最大长度

$$\text{时: } O(n \log n + n) \quad \text{空: } O(n)$$

2043. 简易银行系统

关键：① 判断操作的帐户编号在 $l[n]$ 内
② 转帐、取款时判断帐户内余额是否大于 $money$

时： $O(1)$

空： $O(n)$

606. 根据二叉树创建字符串

先序遍历： 在进行下一次遍历前先加：“(”

在完成一棵子树的遍历后加：“)”

时： $O(n)$

空： $O(\log n)$

2039. 网络的空闲时刻

① 先建图， $g[u].append(v) : g[v].append(u)$

② 广度优先遍历 $ans = \max(ans, (dist \times 2 - 1) // patient[i] \times patient[i] + dist \times 2 + 1)$

时： $O(m+n)$

空： $O(m+n)$

653. 两数之和 IV - 输入 BST

先序遍历的同时用哈希表记录节点的值，若 $k - \text{root.val}$ 在哈希表中，则证明存在两数之和等于 k

时: $O(n)$

空: $O(n)$

2038. 如果相邻的两个颜色均相同则删除当前颜色

统计 Alice, Bob 能删除的次数

若 $Alice \leq Bob$. 则 Bob 胜 ; 否则 Alice 胜 .

时: $O(n)$

空: $O(1)$

440. 字典序的第 K 小数字

字典树思想。

① $\lceil \text{res} \times y, (\text{res}+1) \times y \rceil$ 以 y 为前缀的数字，共 y 个

② y 前缀数和 k 的关系。

若 $\text{cnt} \leq k$ ，说明以 y 前缀的数可跳过，让 res 自增， $k - \text{cnt}$

若 $\text{cnt} > k$ ，说明目标值前缀为 y ，让 $\text{res} \times 10, k - 1$

时: $O(\lg n)$

空: $O(1)$

661. 图片平滑器.

遍历图像 计算每个点周围9个格的平均灰度 (注意图像边界
没有9个周围邻格)

时: $O(m \cdot n)$

空: $O(m \cdot n)$

172. 阶乘后的零

- ① 统计一共有多少对 2、5
- ② 因为 5 的个数少于 2，因此只需统计 5 的个数
- ③ 如 2^5 ，有 2 个 5 的因子， 12^5 有 3 个 5 因子

while n :

$$n // = 5$$

$$\text{ans} += n$$

时: $O(c \log n)$

空: $O(c)$

2028. 找出缺失的观测数据

- ① 求出缺失数据的总和以及均值
- ② 若 均值 < 1 或 > 6 , 返回 $\lceil \rceil$
- ③ 若 $1 \leq$ 均值 ≤ 6 , 若 均值为整数 返回 $\lceil \rceil$ 均值 $\times n$
- ④ 若 均值不为整数 , 添加均值到均值为整数为止

时: $O(n)$

空: $O(n)$

682. 棒球比赛

依据每局的变化 统计 lasttwo, last, cur 的分数，加到 res[]

时: $O(n)$

空: $O(1)$

693. 交替位二进制数

遍历一个数的二进制数，看是否为交替位

时： $O(n)$

空： $O(1)$

2024. 考试的最大困扰度

滑动窗口：

遍历统计 cnt_t 和 cnt_f ，当二者未都大于 k 时，在 k 次修改

内可以使所有答案相同：当二者都大于 k 时，形成窗口。

在窗口内可以使所有答案相同

时： $O(2n)$

空： $O(1)$

1606. 找到处理最多的服务器

busy — 堆排的工作列表

free — 优先队列

res — 统计服务器处理的任务数量

模拟任务到来分配服务器

时: $O(n+k \cdot \log k)$

空: $O(\log k)$

728. 自除数

每个数看能不能被自身任意一位数整除，若存在某位数为0
即不能整除

时: $O(mn)$

空: $O(1)$

954. 二倍数对数组

- ① 数组排序
- ② 哈希表 统计数组中各数次数
- ③ 遍历数组 . 若 $num < 0$, 对应操作 $hashmap[num / 2]$
若 $num = 0$, 看 $hashmap[num]$ 是否为偶数
若 $num > 0$, 对应操作 $hashmap[num \times 2]$

时: $O(n \log n + 2n)$ 空: $O(n)$

420. 强密码检验器.

当长度大于20时，统计连续字符串数量 k ，

当 $k \bmod 3 = 0$ 时，删一个字符串后 $\lfloor \frac{k}{3} \rfloor$ 减 1

当 $k \bmod 3 = 1$ 时，删两个字符串后 $\lfloor \frac{k}{3} \rfloor$ 减 1

$k \bmod 3 = 2$ 时，须删三个字符串后 $\lfloor \frac{k}{3} \rfloor$ 减 1

时： $O(n)$

空： $O(1)$

744. 寻找比目标字母大的最小字母

遍历 letters，若发现有比目标字母大的字母立即返回，否则返回 letters 的第一个字母。

时: $O(n)$

空: $O(1)$

22. 移除元素

遍历 nums，在遇到值为 val 的元素时 pop 出该元素，最后返回数组 nums 的新长度

时: $O(n)$

空: $O(1)$

307. 区域和检索 — 数组可修改

线段树 — 数组表示

$\text{tree} = [0] \times \text{len}(\text{nums}) + \text{nums}$ (偶数下标左节点，奇数下标右节点)

- ① 倒序添加节点 ($n-1, 0, -1$) ; $\text{tree}[i] = \text{tree}[2 \times i] + \text{tree}[2 \times i + 1]$
- ② 修改例) $\text{while } i : \text{tree}[i] += \text{gap}, i // 2$
- ③ 求区间和 $\text{while } l <= r$ · 考虑 l 指向右节点, $\text{res} += \text{tree}[l], l += 1$
· 考虑 r 指向左节点, $\text{res} += \text{tree}[r], r -= 1$

时: $O(\log n)$

空: $O(n)$

762. 二进制表示中质数个计算置位

① 统计一个数二进制的 1 的个数

② 1 的个数是否为质数

时: $O(n \cdot \log n)$

空: $O(1)$

310. 最小高度树

- ① 若 $n=1$ ，直接返回 $[0]$ ；若 $n=2$ ，直接返回 $[0, 1]$
- ② 用 defaultdict 记录所有节点的邻接节点
- ③ 用 deque 记录所有仅有 1 个邻接节点的节点
- ④ bfs，一个节点一个节点地去，当某个点的邻接节点仅有 1 个时，加入 queue

时： $O(n+n)$

空： $O(n)$

796. 旋转字符串

将两串 s 连在一起，看 goal 有没有在 s 中出现

时: $O(n)$

空: $O(n)$

429. N叉树的层序遍历

bfs : 新旧队列广度优先遍历 .

时: $O(n)$

空: $O(c \log n)$

780. 到达终点

- ① 若开始 $sx > tx$ 或 $sy > ty$ ，则不可能到达
- ② 在 $tx > sx$ 和 $ty > sy$ 成立情况下，采用取模的方式快速走
- ③ 走完后看若 x 已经走完的情况下 y 能不能走完

若 y 已经走完的情况下 x 能不能走完

时： $O(\log n)$

空： $O(1)$

804. 唯一摩尔斯密码词.

① 将每个词翻译成摩尔斯密码，若是新的密码，则用哈希表记下来

时: $O(mn)$

空: $O(n)$

357. 统计各位数字都不同的数字个数

① 若 $n=0$ ，返回 1

② 若 $n=1$ ，返回 10

③ 若 $n>2$ ，两位数中的第一位为 1~9 中任一个，第二位为 0~9 中除去第一位的任意一个 ...
即 $9 \times A_9^{n-1}$

时: $O(n)$

空: $O(1)$

8.6. 写字符串需要的行数

rows 代表所需行数， curlen 代表 目前行的长度

遍历字符串，若 $curlen + \text{当前字符长度} \leq 100$ ，加上；否则另起一行

时： $O(n)$

空： $O(1)$

380. $O(1)$ 时间插入、删除和获取随机元素

哈希表 + 数组

插入：加入数组尾部，哈希表记录索引

删除：将要删除的元素换到数组尾部，再进行删除

随机获取：random.randint(0, self.count - 1)

时： $O(1)$

空： $O(2n)$

560. 和为 k 的子数组

前缀和 + 哈希表

遍历计算前缀和并将前缀和用哈希表记录下来，若有 $(k - \text{presum}_i)$ 的相反数存在于哈希表中，则存在和为 k 的子数组

时: $O(n)$

空: $O(n)$

1672. 最富有客户 的资产总量

以行为间隔统计最大子数组和

时: $O(n)$

空: $O(1)$

819. 最常见的单词

- ① 快慢指针 将所有单词的次数用哈希表统计
- ② 哈希表 排序
- ③ 找最高频次且没有被 ban 的词

时: $O(n)$

空: $O(n)$

386. 字典序排序

① 回溯法 开头从 0 开始 不断加数

时: $O(n)$

空: $O(1)$

821. 字符的最短距离

快指针先找到指定字符，再一步步挪慢指针计算最短距离。

时： $O(n)$

空： $O(1)$

388. 文件的最长绝对路径

- ①快慢指针找到一个目录或文件
- ②判断该目录或文件的深度
- ③目录则加入栈中，文件则即时判断绝对路径长度。

时: $O(n)$

空: $O(n)$

824. 山羊拉丁文

- ① 将字符串每个单词分开
- ② 按规定修改每个单词
- ③ 再将单词串成字符串

时: $O(n)$

空: $O(n)$

396. 旋转函数

① 统计数组总和以及初始函数值

② 每一次旋转实际上是函数值 - $\text{nums}[-1] \times n + \text{数组总和}$

时: $O(n)$

空: $O(1)$

868. 二进制间距

快慢指针遍历二进制数 n，统计最大 1 的间距

时: $O(n)$

空: $O(n)$

398. 随机数索引

- ① 哈希表 将所有数的索引都 记录下来
- ② 再在对应目标数的数组 中随机挑一个索引 输出

- ① 字典的值可以是一个列表
- ② `random.choice(list)` 从列表中随机挑一个元素

时: $O(n)$

空: $O(n)$

883. 三维形体投影面积

- ① xy 面投影看 grid 中非 0 元素个数
- ② xz 面投影看 grid 中每行最大元素之和
- ③ yz 面投影看 grid 中每列最大元素之和.

时: $O(n)$

空: $O(1)$

427. 建立四叉树

递归将矩阵分上、上左、上右、下左、下右 5 部分，如果出现矩阵所有元素都相等则返回，否则将继续递归分矩阵

时： $O(n^2 \log n)$

空： $O(1)$

908. 最小差值 I

- ① 找到数组中最大值与最小值
- ② 若差值小于等于 2 倍 k，则返回 0
- ③ 若差值大于 2 倍 k，则返回 $\text{最大值} - \text{最小值} - 2 \times k$

时: $O(n)$

空: $O(1)$

1305. 两棵二叉搜索树中的所有元素

① 先用列表将第一棵树的元素按中序遍历记录下来

② 第二棵树在进行中序遍历的时候再对比列表中的元素，添加到 res_p

时: $O(2n)$

空: $O(n)$

937. 重新排列日志文件

自定义排序 logs.sort(key=trans)

```
def trans(log):
```

```
a, b = log.split(' ', 1)
```

分解的次数

时: $\mathcal{O}(n \log n)$

空: $\mathcal{O}(1)$

9.33. 最近的请求次数

- ① 按顺序在 `self.queue` 里记录 ping 的时刻
- ② 每次添加前先去掉 时刻 小于 $t - 3000$ 的 Ping .

时: $O(n)$

空: $O(n)$

433. 最小基因变化

根据 bank 用广度优先遍历看 start 看哪种变异的结果

时: $O(n)$

空: $O(n)$

442. 数组中重复的元素

遍历数组，将 $\text{nums}[\text{abs}(\text{nums}[i])]$ 做记号。若某次在做记号前发现该位置已经有记号了，则说明 $\text{nums}[i]$ 是重复元素。

时: $O(n)$

空: $O(1)$

942. 增减字符串匹配

① $small = 0$, $big = \text{len}(s)$

② 遍历字符串。若为 "I", 则 res 增加 small, $small + 1$;

若为 "D", 则 res 增加 big, $big - 1$;

③ 最后 res 再添 small

时: $O(n)$

空: $O(1)$

449. 序列化和反序列化二叉搜索树

① 序列化：后序遍历再连成字符串

② 深度优先遍历：1. 右根左顺序

2. 若 vals 为空 或 $\text{vals}[i] < \text{lower}$ 或 $\text{vals}[i] > \text{upper}$

return None

否则 $\text{root.right} = \text{dfs}(\text{val}, \text{upper})$

$\text{root.left} = \text{dfs}(\text{lower}, \text{val})$

时： $O(n)$

空： $O(n)$

944. 打乱数组

小列为单位遍历数组，若发现有当前字符序小于前一个字符，则停止遍历该列，且 $\text{res} + 1$

时: $O(n)$

空: $O(1)$

面试题 01.05. 一次编辑

- ① 若 $\text{first} = \text{second}$, 返回 0
- ② 若 first 与 second 长度差大于 1 , 返回 -1
- ③ 若 first 与 second 长度一致 , 遍历看有几处不同 .
- ④ 若 first 比 second 短 , 遍历看是否添一个元素使二者一致
- ⑤ 若 first 比 second 长 , 遍历看是否删一个元素使二者一致

时: $O(n)$

空: $O(1)$

691. 贴纸拼词

dfs 同时记录状态：用某个词拼过之后在队列中加入 (next , $\text{step}+1$)

时: $O(n)$

空: $O(n)$

812. 最大三角形面积

海伦公式 : $P = \frac{a+b+c}{2}$ $S = \sqrt{P(P-a)(P-b)(P-c)}$

遍历所有点，先判断是否能构成三角形，再计算面积

时: $O(n^3)$

空 $O(1)$

953.验证外星语词典

遍历words数组，看每个word是否按外星语字典序排序

- 1.若前后两个word完全一致，跳过
- 2.若前词字典序大于后词，返回False
- 3.后词序大于前词，则检测下一组词

时: $O(mn)$

空: $O(1)$

668. 乘法表中第 k 小的数

- ① 二分查找确定范围
- ② 然后通过乘法表有序性快速计算元素个数

时: $O(m \log n)$

空: $O(1)$

462. 最少移动次数使数组元素相等 II.

① 将数组排序

② 遍历求所有数到数组最中间的数的移动次数总和.

时: $O(n \log n)$

空: $O(1)$

436. 寻找右区间

- ① 在每个区间后加上自己的索引下标
- ② 将区间按 start 排序
- ③ 用二分查找 找到当前区间的最小右区间

时: $O(n \log n)$

空: $O(1)$

961. 在长度 $2N$ 的数组中找出重复 N 次的元素

① 随机取2个数，看他们是否相等

时: $O(1)$

空: $O(1)$

② 重复元素之间间隔最大是 3 , 看间隔 1、2、3 有无相同元素

时: $O(n)$

空: $O(1)$

4/4. 我能赢吗

- ① 若可选数中有大于目标和者，先手直接胜
- ② 若可选数和不大于目标和，没有胜方
- ③ 在选数时，先手方若在自己做决策后大于目标和或者对方未能有大于目标和，该数状态记为 True，否则记为 False。

时: $O(2^n)$

空: $O(2^n)$

675. 为高尔夫比赛砍树

- ① 将所有树排序，并记录树的坐标
- ② 每两棵树采用 BFS 方法 计算 步长
- ③ 所有步长求和

时: $O(n^2)$

空: $O(mn)$

965. 单值二叉树

层序遍历，看二叉树的每个值是否都是同一个值

时： $O(n)$

空： $O(\log n)$

467. 环绕字符串中唯一的子字符串

- dp :
- ① 若前后两字符是邻居字符，则 $k+1$ ，否则 $k = 1$
 - ② 每次更新 ①之后 更新 $dp[\text{当前字符}] = \max(dp[\text{当前字符}], k)$
 - ③ 最后求 dp 里总和

时: $O(n)$

空: $O(1)$

699.掉落的方块

- 1.用drop数组记录已落方块的（始点，末点，高度）
2. 每落下下一个方块，遍历已落方块，通过方块始末坐标以及高度判断当前方块要落的高度
3. 每落一个方块要更新res数组和当前的最大高度maxh

时: $O(n \times n)$

空: $O(n)$

面试题. 17.11. 单词距离

遍历所有单词，当遇上 word₁ 或 word₂ 时，若之前未记录过 last word，则当前 word 为 last word，在后期重复遇到 last word，则更新下标，若遇到另一个 word，则算单词距离，并更新 last word 为当前 word

时: $O(n)$

空: $O(1)$

1021. ~~删除~~ 最外层括号

遍历得到每一组完整的括号，把每一组括号的最外层括号去掉后
添加到 res

时: $O(n)$

空: $O(n)$

468. 验证IP地址

- ① 将地址以“：“和以“.”分开
- ② 若分了之后长度为8按IPv6标准检查，长度为4按IPv4标准检查
- ③ IPv6 标准：

长度在 1~4	/	字母只在 a~f, A~F
不得有符号		
- ④ IPv4 标准：

必须是数字	/	在 0~255 之间
不得多个 0 开头		

时: $O(n)$

空: $O(1)$

1022. 从根到叶的二进制之和

中序遍历，把从根到叶的二进制数表示出来，再进行转换
成十进制求和

时： $O(n)$

空： $O(\log n)$

525. 连续数组

- ① 遍历计算前缀和 pre，若当前数为 0 $pre \leftarrow -1$ ，若为 1 $pre \leftarrow +1$ 。
- ② 若 $pre = 0$ ，则记录 当前下标 $res = i + 1$
- ③ 若 $pre \neq 0$ ，且 pre 不在 hashmap， $hashmap[pre] = i$
- ④ 若 $pre \neq 0$ ，且 pre 在 hashmap， $res = \max(res, i - hashmap[pre])$

时: $O(n)$

空: $O(n)$

473.火柴拼正方形

回溯法： 1. 若火柴总长不是4的倍数， 返回false
2. 将火柴按长度倒序排序
3. 将火柴分配给每条边， 若分配给该边长度小于等于火柴总长的四分之一即可

时： $O(4^n)$

空： $O(4)$

450. 删除二叉搜索树中的节点

- 分情况：① root 为空 或找不到 key 值节点， return root
- ② key 值节点为叶子节点， 删掉该节点
- ③ key 值节点为非叶节点， 左子树代替该节点， 右子树成为
左子树的最右节点 的右孩子

时： $O(h)$

空： $O(1)$

829. 连续整数求和

根据等差数列求和公式 $\frac{(a+a+k-1) \times k}{2} = n \Rightarrow \frac{2n}{k} - k + 1 = 2a$

a 和 k 均为正整数 , $\frac{2n}{k} \geq k+1 \Rightarrow \frac{2n}{k} > k$

在 $[1, \sqrt{2n}]$ 枚举 k , 若 k 为 $2n$ 约数 , 则 (a, k) 合法

时: $O(\sqrt{n})$

空: $O(1)$

929. 独特的电子邮件地址

- ① 对于每个地址用 `split('@')` 分开本地名和域名
- ② 本地名中 ‘.’ 忽略， ‘+’ 后的内容删掉
- ③ 域名则不作修改
- ④ 用哈希表记录处理过后的邮件地址

时: $O(mn)$

空: $O(mn)$

478. 在圆内随机生成点

通过随机函数 `random.uniform(a, b)` 生成点，只要生成的点在
给定圆的范围内即可

时: $O(1)$

空: $O(1)$

732. 我的日程安排表

差分数组记录从小到大时刻，若该时刻为 start，则 +1，若为 end，则 -1，再遍历一遍差分数组记录到该时刻为止最大的预约次数

时: $O(n^2)$

空: $O(n)$

875. 爱吃香蕉的珂珂

二分查找， $i = 1$ ， $j = \max(piles)$

$$mid = (i+j) / 2$$

若每轮吃 mid 根 轮数大于 h ，则 $i = mid + 1$

若 小于 h ，则 $j = mid - 1$

时： $O(n \log m)$

空： $O(1)$

1037. 有效的回力镖

① 不能有一样的点

② 两两斜率不能一样

时: $O_{(1)}$

空: $O_{(1)}$

497. 非重叠矩形中的随机点.

根据矩形中包含的整数点的个数分配选中的概率，以此保证
选中每个点都是等概率的

时: $O(n)$

定: $O(n)$

890. 查找和替换模式

- ① 小头遍历字符串，用哈希表记录第几个新字符，然后用等长数组记录该字符串对应哈希表的值
- ② 模式串和单词串都进行①操作
- ③ 返回那些①结果与模式串相同的单词串

时: $O(mn)$

空: $O(n)$

105|. 高度检查器

- ① $\text{heights}'$ 为 heights 排好序的版本
- ② 对比 $\text{heights}'$ 与 heights ，看有几个不同的数

时: $O(n \log n)$

空: $O(n)$

498. 对角线遍历

① 往上则 $x-1, y+1$; 往下则 $x+1, y-1$

② 遇到边界需要特殊处理

时: $O(nm)$

空: $O(1)$

532. 数组中的 k-diff 数对

- ① 一个哈希表记录出现过的数字，一个哈希表记录匹配的数对
- ② 从头遍历数组，把出现过的数字记录下来，并把头一次出现的符合 k-diff 的数对记录下来

时: $O(n)$

空: $O(2n)$

719. 找出第 k 小的数对距离

① 数组排序

② 二分法确定数对距离 dis

③ 给定 dis 用双指针法计算小于等于 dis 的数对个数

时: $O(n \log n + n)$

空: $O(1)$

108. 复写零

① 从头开始遍历

② 若遇到 0 则复写，并且 pop 掉 数组最后一个元素

时: $O(n)$

空: $O(1)$

剑指 offer 11. 029 . 排序的循环链表

① 把目标节点生成出来 target

② 遍历一遍链表找出目标节点位置 cur

(1) $cur.val = target.val$

(2) $cur.val < target.val$ 且 $cur.next.val > target.val$

(3) 当 $cur.val > cur.next.val$ 时

1. $cur.val < target.val$

2. $cur.next.val > target.val$

都在 cur 后接上 target

时: $O(n)$

空: $O(1)$

508. 出现次数最多的子树元素和

后序遍历，记录包括根节点在内的所有子树和，并用哈希表记录下来，最后从哈希表中找到出现次数最多的子树和

时: $O(n)$

空: $O(n)$

1108. IP 地址无效化

将 IPv4 地址中的“.” 替换成 “[.]”

时: $O(n)$ 空: $O(1)$

513. 找树左下角的值

层序遍历，在每层遍历之前记录下最左边的值

时： $O(n)$

空： $O(\log n)$

30. 串联所有单词子串

- ① 哈希表记录单词子串出现的次数 $hashmap$
- ② 通过滑动窗口遍历字符串 s ，并用 $temp$ 哈希表记录子串出现次数
- ③ 若 $temp = hashmap$ ，记录起始位置 i

时: $O(mn)$

空: $O(n)$

剑指 offer 11 091. 粉刷房子

动态规划：维护 3×2 的数组，第 1 列是上一个房子粉刷价格，

每次挑中除自身油漆外最便宜的价格

时： $O(n)$

空： $O(1)$

515. 在每个树行中找最大值

层序遍历，每层遍历前先记录该层最大值

时: $O(n)$

空: $O(\log n)$

710. 黑名单中的随机数

- ① 记录白名单里数字的个数 m
- ② 维护白名单字典从 mnn ，若某个黑名单数 $\geq m$ ，则从白名单中去掉
- ③ 维护候选名单字典，若某个黑名单数 $< m$ ，则候选名单[黑名单数] = 白名单字典.pop()

时: $O(2n)$

空: $O(m)$

5.22. 最长特殊序列 ||

题意：特殊子序列不能是其他字符串的字序列

逐个对比，某个字符串是否是其他字符串的子序列

时间： $O(nm)$

空间： $O(m)$

324. 摆动排序 II

- ① 先将数组排序
- ② 数组前半部分倒序，后半部分倒序
- ③ 按前半部分、后半部分次序向数组前插入排序

时: $O(n \log n + n)$ 空: $O(1)$

535. Tiny URL 的加密与解密

- ① 用哈希表任意存一个与 URL 映射的 Tiny URL
- ② 当获取 TinyURL 可从 哈希表中取出原 URL

时: $O(1)$

空: $O(n)$

1175. 质数排列

- ① 找出 $1 \sim n$ 的质数个数 a ，和非质数个数 b
- ② 排列组合 $(a! \times b!) \bmod (10^9 + 7)$

时: $\mathcal{O}(n \cdot m)$

空: $\mathcal{O}(1)$

871. 最低加油次数

从0~n次加油机会，看给定最低加油次数能否跑到终点。

时: $O(n^2)$

空: $O(1)$

556. 下一个更大的元素 III

倒序遍历数位，若对于某个数位，在其右边有比其更大的数，则从中选取最小的比该数位大的数，与该数位交换后，右边所有数位排序，最后得到的数就是下一个更大的元素。

时： $O(n^2)$

空： $O(1)$

1200. 最小绝对差

① 整数数组排序

② 记录 相邻的两个数最小的绝对差

时: $O(n \log n + n)$ 空: $O(1)$

729. 我的日程表安排 |

- ① 哈希表记录已安排的日程
- ② 在得到一个新的日程的时候，遍历哈希表看该日程是否冲突，不冲突则加入哈希表。

时: $O(n^2)$

空: $O(n)$

648. 单词替换

- ① 在给定的词根词典构造字典树
- ② 句子按空格分开
- ③ 每个词都尝试能否找到最短词根

时: $O(n+m)$

空: $O(n)$

741. 摘樱桃

- ① 先用 dp 计算出从左上到右下的最大樱桃数
- ② 倒序遍历走一遍最大樱桃数的路线，把途中有樱桃的清空
- ③ 倒序 dp 计算从右下回到左上最大樱桃数

时: $O(3n^2)$

空: $O(n^2)$

676. 实现一个魔法字典

对字典每个单词与所给单词逐一字符比较，差别为1则 True

时: $O(nm)$

实 $O(n)$

745. 前缀和后缀搜索

- ① $\text{temp} = \text{word} + "\#" + \text{word}$ 可搜索的后缀+前缀
- ② 后缀和前缀可为空，从 $0 \sim \text{len}(\text{word}) + 1$ 遍历，构建字典树
- ③ 搜索时， $\text{suff} + "\#" + \text{pref}$ 来搜索

时: $O(nm)$

空: $O(nm)$

735. 行星碰撞

- ① 从头往后遍历，如遇 $|i| > 0$ 且 $|j+i| < 0$ ，则 继续
- ② 若 $|i| = |j+i|$ ，二者都去掉
- ③ 否，去掉绝对值更小的

时： $O(n)$

空： $O(1)$

1252. 奇数值单元格的数目

- ① 建长度为 m 和 n 的数组，表示长和宽
- ② 遍历 indices， $\text{indices}[i][0]$ 则在对应 m 数组 +1， $\text{indices}[i][1][j]$ 在对应 n 数组 +1
- ③ 统计 m、n 数组奇数偶数的个数
- ④ 答案为 m、n 数组奇、偶个数相乘相加

时: $O(m+n+p)$

空: $O(m+n)$

剑指 offer 11 041. 滑动窗口的平均值

① 若滑动窗口未满，直接往里添数后求平均

② 若滑动窗口已满，按先进先出的顺序弹出一个数再添数求平均

时: $\mathcal{O}(1)$

空: $\mathcal{O}(n)$

5.8 四叉树交集

递归的思想：若两个四叉树的根有其中之一是叶子节点：

①若其值为True，则返回 Node(True, True)

②其值为False，返回另一四叉树

否则，递归计算四个子节点的交集：

①若子节点均为叶子节点且值相同，返回 Node(o1.val, True)

否则返回 Node(False, False, o1, o2, o3, o4)

时： $O(mn)$

空间： $O(\log mn)$

565. 数组嵌套

因为元素为 $0 \sim n-1$ ，所以所有元素会形成一个或多个环

① 从 $0 \sim n-1$ 开始遍历元素

② 遍历过的元素标记为 n ，若遇到 n 则意味着碰到环了。

记录下环的大小。

时: $O(n)$

空: $O(1)$

749. 隔离病毒

- ① 找到每个病毒区域，并计算有多少未感染的邻居
- ② 拥有最多的邻居的病毒区域包围防火墙，墙数等于邻居数
- ③ 其它病毒区域蔓延一格
- ④ 继续上述操作直至没有病毒区域或没有未感染的邻居

时: $O(mn)$

空: $O(m+n)$

1260. 二维网格迁移

① 移动次数 $k = k \% (m \times n)$

② 每次移动直接把每行最后一个元素挪到下一行
的首个元素，最后一行则挪到第一行

时: $O(m \cdot k)$

空: $O(1)$

731. 我的日程安排表 II

- ① 根据所给的时间段建立线段树，节点值为访问的次数
- ② 查询，若节点值小于 2 则可以预订

时: $O(mn)$

空: $O(m)$

757. 设置交集大小至少为 2

- ① 将 intervals 按 start 顺序、end 逆序排序
- ② res 先添加 intervals 最后一个区间的 start+1，start
- ③ 从右往前遍历 intervals，有 3 种情况：
 1. $s \leq res[-1] \leq e$ 且 $s \leq res[-2] \leq e$ ，略过
 2. $s \leq res[-1] \leq e$ ，添加 s 入 res
 3. 1. 2 都不成立，添加 $s+1, s$ 入 res

时: $O(n \log n + n)$

空: $O(n)$

814. 二叉树剪枝

- ① 后序遍历，如遇空节点，返回值 0
- ② 每个节点返回左孩子 + 右孩子 + 自身的值
- ③ 若某节点左右孩子值为 0，则剪枝

时: $O(n)$

空: $O(c \log n)$

剑指 Offer II. 115. 重建序列

- ① 统计好每个节点的下一个可选邻居
- ② 统计每个节点的入度
—— — — — — —
- ③ 从入度为 0 的节点开始遍历，若任何时候 入度为 0 的节点大于 1，则说明超序列不唯一，小于 1 则说明没有超序列

时: $O(mn)$

空: $O(mn)$

9/9. 完全二叉树插入器

- ① 统计现有树的节点数
- ② 根据完全二叉树的特性找到新节点的父节点
- ③ 添加新节点

时: $O(\log n)$

空: $O(1)$

1184. 公交站间的距离

以起点开始，两个方向同时遍历，到达终点则停止，直至双方都到达终点，返回距离较短的

时： $O(n)$

空： $O(1)$

1206. 设计跳表

- ① 链表节点： next 和 down 和 val
- ② 跳表初始化： 16 层有序单链表 且高层 down 指向下一层，
同层 next 指向下一元素，首层为 head
- ③ 查找： 从首层开始，逐层向下查找，找到目标元素为 True，否则 False
- ④ 插入： 在每层找到插入位置，最底层插入之后，有 $\frac{1}{2}$ 概率选择
是否在上层插入，插入后 down 指向下层
- ⑤ 删除： 从首层开始删，把每一层含有目标的删去

时: $O(\log n)$

空: $O(n)$

592. 分数加减运算

① 初始分数为 $\frac{0}{1}$

② 逐个读取分数进行相加减

③ 最后 gcd 函数约分

时: $O(n)$

空: $O(1)$

1331. 数组序号转换

- ① 数组排序
- ② 哈希表 将每个数的顺序记下，相同的数忽略
- ③ 按哈希表记录的顺序改原数组上元素，替换成对应的顺序

时: $O(n \log n + n)$ 空: $O(n)$

593. 有效的正方形

- ① 统计两两点之间的距离，若有两点距离为 0 则返回 False
- ② 若统计的距离中，出现 4 组是一样的。另外 2 组是一样的，则返回 True。否则返回 False

时: $O(n)$

空: $O(n)$

952. 按公因数计算最大组件大小

- ① 并查集可以统计出属于同一个组件的数
- ② 在所有数都进行合并之后，遍历数组 `nums`，对于每个数得到其所在的组件并更新该组件的大小

时： $O(n \times \alpha(n) \times \sqrt{m})$ 空： $O(m)$

1161. 最大层内元素和

层序遍历，计算每层的元素和，并且记录元素和最大

的那一层号

时： $O(n)$

空： $O(\log n)$

1374. 生成每种字符都是奇数个的字符串

① 若 n 为奇数，返回 n 个 'a'

② 若 n 为偶数，返回 n-1 个 'a' 和 1 个 'b'

时: $O(n)$

空: $O(1)$

6.22. 设计循环队列

记录目前队列的首、尾以及已有元素个数，若未满可在队尾
添加，若不空可以在队首出队

时: $O(1)$

空: $O(n)$

899. 有序队列

① 若 $k=1$ ，则每次把串首字符挪到串尾，记录下字典序最小的字符串

② 若 $k>1$ ，则必能得到将字符串按字典序排序的序列

时： $O(n \log n)$

空： $O(n)$

1403. 非递增顺序的最小子序列

- ① 数组逆序排序
- ② 排选最大的元素直至求和大于原数组的总和.

时: $O(n \log n + n)$

空: $O(n \log n)$

623. 在二叉树中增加一行

- ① 先采用层序遍历 找到 $\text{depth} - 1$ 行的节点
- ② 左节点接新节点，新节点的左子节点接原来的左子节点。
右子节点也同样操作
- ③ 若 $\text{depth} = 1$ ，直接将 root 接在新节点的左子节点上

时: $O(n)$

空: $O(1)$

761. 特殊的二进制序列

把“1”当作左括号，把“0”当作右括号，最终要让前面左括号尽可能多

- ① 特殊序列一定是以“1”开头，“0”结尾
- ② 把当前的特殊序列排成最大，递归去掉头和尾
- ③ 将特殊序列降序排列后拼接

时: $O(n^2)$

空: $O(n^2)$

1413. 逐步求和得到正数的最小值

① 逐步求和找到最下限的值

② 若下限大于0，则返回1

③ 若下限小于等于0，则返回下限的绝对值加1

时: $O(n)$

空: $O(1)$

640. 求解方程

① 先将等号左右分开

② 若表达式开头为“-”，则在开头加“ $= 0$ ”

③ 将表达式中逐个变量分开，配合操作符号让未知变量 x 和已知数合并

时： $O(n)$

空： $O(n)$

1417. 重新格式化字符串

- ① 将数字与字母分开 并统计数量
- ② 若数量差大于 1 则不可被格式化
- ③ 按规则将字符串格式化

时: $O(n)$

空: $O(n)$

1282. 用户分组

- ① 用哈希表统计出处于不同分组大小的下标
- ② 若处于该分组大小的数量恰好等于该分组大小，则直接加入 res 中
- ③ 否则，需要分成若干份该分组大小的数组分别加入 res 中

时: $O(n+n)$

空: $O(n)$

1450. 在既定时间做作业的学生人数

从头遍历每个学生，看 querytime 是否在 startime[i] 和 endtime[i]
之间

时: $O(n)$

空: $O(1)$