

# 力扣心得

---

2026-03-30

## 两数之和

---

- 哈希表
- 暴力枚举

## 字母异位词分组

---

哈希表 + `sort`

## 最长连续序列

---

无序集合，实时更新最大值：将数字存到集合，找到连续序列的最小值，从这个值开始循环查找当前值的下一个数，统计序列长度，最后进行更新

## 移动零

---

双指针：一个指针指向非零元素序列的下一个位置，用于存储下一个非零元素，另一个指针从头到尾遍历一次数组，移动非零元素

## 盛最多水的容器

---

双指针、短板效应

## 三数之和

---

三指针、排序：记得跳过重复元素

## 接雨水

---

动态规划：维护两个数组，记录当前柱子左边的最高点和右边的最高点，用短板效应可求出当前柱子可接到的雨水

## 无重复字符的最长子串

---

## 找到字符串中所有字母异位词

---

滑动窗口：维护两个cnt数组，使用 `vector` 对 `==` 的重载加速判断

## 和为 K 的子数组

---

前缀和、哈希表：通过哈希表进行O(1)查找，在哈希表中维护前缀和及其出现次数进行快速判断

## 滑动窗口最大值

---

单调栈：栈存最大值的下标

## 最小覆盖子串

---

哈希表、滑动窗口、双指针：哈希表统计盈亏字符数，维护一个变量统计还需要多少字符数

- 不缺字符，左指针收缩，找最小子串
- 缺字符，右指针扩张，找子串

## 最大子数组和

---

动态规划：dp代表从当前位置往前的最大子数组和

- 如果前缀和加上当前元素比直接使用当前元素还小，直接抛弃前缀和

前缀和：维护当前位置的前缀和与当前位置前缀和的最小值

## 合并区间

---

模拟、自定义排序

## 轮转数组

---

三次翻转

## 除了自身以外数组的乘积

---

维护左右乘积列表

可优化为 $O(1)$ 空间复杂度

## 缺失的第一个正数

---

## 矩阵置零

---

维护标记数组

## 螺旋矩阵

---

维护边缘变量、模拟

## 旋转图像

---

原地旋转：后续补充

## 搜索二维矩阵 II

---

利用数据特性快速查找

## 相交链表

---

双指针：两个指针完整走过两个链表，最后肯定会相交或者同时为null

## 反转链表

---

模拟

## 回文链表

---

额外数组存储节点值

## 环形链表

---

快慢指针

## 环形链表 II

---

快慢指针

## 合并两个有序链表

---

模拟

## 两数相加

---

模拟

## 删除链表的倒数第 N 个结点

---

双指针：一个指针先走n步，然后两个指针一起移动，直到先走的指针走到链表末尾

## 两两交换链表中的节点

---

迭代、模拟

## K 个一组翻转链表

---

模拟

## 随机链表的复制

---

哈希表：哈希表存储原节点和最终节点的映射

## 排序链表

---

递归、归并排序

## 合并 K 个升序链表

---

优先队列

## LRU 缓存

---

我用go写的，使用虚拟头节点可以避免特判很多特殊情况

## 二叉树的中序遍历

---

dfs即可

## 二叉树的最大深度

---

dfs: 递归函数返回当前节点的最大深度

## 翻转二叉树

---

dfs: 递归函数对当前节点的左右子树进行翻转

- 先递归再翻转当前节点子树

## 对称二叉树

---

bfs: 层序遍历

## 二叉树的直径

---

dfs: 递归函数返回当前以当前节点为根节点的树的直径，实时更新最大值

## 二叉树的层序遍历

---

bfs: 层序遍历

## 将有序数组转换为二叉搜索树

---

递归: 后续补充

## 验证二叉搜索树

---

dfs: 递归函数返回当前节点是否为有效二叉搜索树，递归函数要传递最小、最大值

- 最小、最大值取决于节点所在位置

## 二叉搜索树中第 K 小的元素

---

中序遍历

## 二叉树的右视图

---

bfs: 注意节点插入顺序

## 二叉树展开为链表

---

模拟: 将当前节点左子树接到当前节点右子树

## 从前序与中序遍历序列构造二叉树

---

递归: 从前序遍历数组寻找根节点，在中序遍历数组分离左右子树，递归时重新构建前序、中序遍历数组

## 路径总和 III

---

哈希表、前缀和、dfs、回溯：自己试着敲

## 二叉树的最近公共祖先

---

dfs: 递归函数返回当前节点找到的目标节点或者他们的最近公共祖先

## 二叉树中的最大路径和

---

dfs: 动态更新最大值，递归函数返回当前节点子树的最大路径和（只能选取左子树或者右子树）

## 岛屿数量

---

bfs: 对每个点都使用一次广搜，同时更新遍历标记

## 腐烂的橙子

---

bfs: 需要维护变量标记当前层是否有橘子腐烂

## 课程表

---

## 实现 Trie (前缀树)

---

每个节点存一个字符，节点可复用

## 全排列

---

回溯

## 子集

---

回溯