

数据结构与算法

学习方法

- 普用工具
- 刻意练习
- "五毒神掌"

时间、空间复杂度

- $O(1)$
- $O(N)$
- $O(N^2)$
- $O(2^N)$
- $O(\log N)$
- $O(k \log N)$
- $O(N!)$

数组

- 连续的内存空间
- 查找 $O(1)$
- 插入、删除 $O(N)$

链表

- 单链表
- 双链表
- 查找 $O(N)$
- 插入、删除 $O(1)$
- 头结点

栈

- 先入后出
- 链表实现

队列

- 普通队列 先入先出
- 双端队列 双端都可进出
- 优先级队列 正常入，按优先级出
- 链表、数组实现

树

- 二叉树
 - 前序遍历
 - 中序遍历
 - 后序遍历
- 二叉搜索树
 - $O(\log N)$
 - 中序遍历：递增排序排列
 - 特点
 - 左子树均小于根节点
 - 右子树均大于根节点
 - 每个节点均是二叉搜索树
- 面试题解法一般是递归
- 空间换时间
 - Trie树 字符串匹配
 - 建树
- AVL树
- 红黑树

位运算

- 计算机底层，速度快
- 取代除法
- 熟记基本解法

动态规划

- 递推
- 状态的定义
- 最优子结构
- 状态转移方程
- 剪枝

贪心算法

- 子问题
- 子问题最优解是全局最优解
- 找到合适的场景
- 不能回退
- 每次最优，不一定全局最优

遍历和搜索

- $O(N)$ 深度优先DFS
- $O(N)$ 广度优先BFS
- 非递归：栈、队列
- 实现 BFS、DFS

分治

- 找出重复子问题
- 多用递归的方式解决问题
- 缓存子问题

递归

- 解法
 - 1. 终止条件
 - 2. 处理当前层逻辑，设置状态
 - 3. 下探(可能多次)
 - 4. 反转状态

list, map, set

- 有序
 - $O(\log N)$ treeSet
- 无序
 - 数据量大时更快
 - $O(1)$ hashSet
 - 退化-链表 $O(N)$