



▼ 算法
▼ 复杂度
▼ 时间复杂度
▼ O(1) 最佳
常数复杂度
<ul style="list-style-type: none">Hash Table
<ul style="list-style-type: none">Cache
▼ O(Log n)
仅次常数复杂度
<ul style="list-style-type: none">二分查找
<ul style="list-style-type: none">二叉搜索树
<ul style="list-style-type: none">O(n)
线性复杂度， 如大多数遍历操作
<ul style="list-style-type: none">O(n^2)
双重 for 循环
<ul style="list-style-type: none">O(2 ^ n)
递归的时间复杂度
▼ 空间复杂度
<ul style="list-style-type: none">O(1)
原地操作
<ul style="list-style-type: none">O(n)
开辟线性辅助空间
▼ 数组
<ul style="list-style-type: none">连续空间
<ul style="list-style-type: none">查找快、插入/删除结点慢
▼ 链表
<ul style="list-style-type: none">离散空间
<ul style="list-style-type: none">查找慢、插入删除结点快

<div><div><div><div><div></div><div>栈</div></div></div><div><div></div><div>先进后出</div></div></div></div>
<div><div><div><div><div></div><div>队列</div></div></div><div><div></div><div>先进先出</div></div></div></div>
<div><div><div><div><div></div><div>映射</div></div></div><div><div></div><div>K/V</div></div></div></div>
<div><div><div><div><div>▼</div><div>集合</div></div></div><div><div></div><div>Key 不重复</div></div></div></div>
<div><div><div><div><div></div><div>🚩 集合内的元素不重复， 不存在Key的概念</div></div></div></div></div>
<div><div><div><div><div>▼</div><div>★ 并查集</div></div></div></div></div>
<div><div><div><div><div></div><div>站队问题</div></div></div></div></div>
<div><div><div><div><div></div><div>初始化</div></div></div></div></div>
<div><div><div><div><div></div><div>查询合并</div></div></div></div></div>
<div><div><div><div><div></div><div>路径压缩</div></div></div></div></div>
<div><div><div><div><div>▼</div><div>树</div></div></div></div></div>
<div><div><div><div><div>▼</div><div>★ 二叉树</div></div></div></div></div>
<div><div><div><div><div>▼</div><div>遍历</div></div></div></div></div>
<div><div><div><div><div></div><div>DFS</div></div></div></div></div>
<div><div><div><div><div></div><div>BFS</div></div></div></div></div>
<div><div><div><div><div>▼</div><div>平衡二叉树</div></div></div></div></div>
<div><div><div><div><div></div><div>AVL 树</div></div></div></div></div>
<div><div><div><div><div></div><div>红黑树</div></div></div></div></div>
<div><div><div><div><div></div><div>剪枝</div></div></div></div></div>
<div><div><div><div><div></div><div>★ 字典树</div></div></div><div><div></div><div>空间换时间</div></div></div></div>
<div><div><div><div><div>▼</div><div>图</div></div></div></div></div>
<div><div><div><div><div></div><div>遍历时需要记录访问过的结点</div></div></div></div></div>

▼ 递归、分治

- 🌟 盗梦空间

- 终止状态

- 本层处理

- Drill Down

- 分层状态清理

▼ 二分查找

- 有序

- 有界

- 能够通过索引随机访问

▼ 贪心算法

- 判断能不能贪心

- 弱化版的动态规划

▼ 动态规划

- 高级版本是递推公式

- 简单版本是递归+缓存

- 状态的定义

- 最优化结构

- 状态转移方程

▼ 位运算

- ❤️ 常见的位运算公式， 记忆

▼ 布隆过滤器

- 判断不存在 100% 准确

- 判断存在误差

- 利用Hash函数将判断 Kewy 对应到多个位上

▼ LRU

- HashTable + 双向链表
- get 和 set 都是 $O(1)$ 复杂度