

Algorithm

算法 Algorithm

- 递归 Recursion
- 贪心 Greedy
- 分治 Divide And Conquer
- 回溯 Backtrace
- 动态规划 Dynamic Programming

排序 Sort

- $O(n^2)$
 - 插入排序 Insertion Sort
 - 希尔排序 Shell's Sort
 - 冒泡排序 Bubble Sort
 - 选择排序 Selection Sort
 - $O(n \log n)$
 - 快速排序 Quick Sort
 - 归并排序 Merge Sort
 - 堆排序 Heap Sort
 - $O(n)$
 - 桶排序 Bucket Sort
 - 计数排序 Counting Sort
 - 基数排序 Radix Sort
- 改进
- 更受欢迎
- 实际场景很少用，用于学习
- 线性排序，使用条件苛刻
- 关注点
- 执行效率
 - 最好、最差、平均时间复杂度
 - 时间复杂度的系数、常数、低阶
 - 比较和交换次数
 - 内存消耗
 - 是否原地排序，即空间复杂度 $O(1)$
 - 稳定性
 - 是否是稳定排序算法，及等值元素相对位置不变
 - 概念
 - 满有序度 $n(n-1)/2$
 - 有序度，具有有序的元素个数
 - 逆序度，等于满有序度 - 有序度

查找 Search

- $O(\log n)$
- 二分查找 Binary Search

字符串匹配

- BM 算法
- KMP 算法
- RK 算法
- BF 算法
- Aho-Corasick 算法 (AC 自动机)
- Trie 树 (字典树)

位操作

- 布隆过滤器 Bloom Filter

其他算法

- LRU Cache
- A* 寻路算法

复杂度分析

- 分析
 - 最好
 - 最差
 - 平均
 - 均摊
- 时间复杂度
 - Big O Notation
 - $O(1)$ Constant Complexity 常数时间复杂度
 - $O(\log n)$ Logarithmic Complexity 对数时间复杂度
 - $O(n)$ Linear Complexity 线性时间复杂度
 - $O(n \log n)$
 - $O(n^2)$ Square Complexity 平方时间复杂度
 - $O(n^3)$ Cube Complexity 立方时间复杂度
 - $O(2^n)$ Exponential Growth 指数时间复杂度
 - $O(n!)$ Factorial 阶乘时间复杂度
- 空间复杂度
 - 额外使用的空间
 - $O(1)$ 原地操作
 - $O(n)$ 线性空间

数据结构 Data Structure

数组 Array

- 连续内存空间
- 按数组下标访问 $O(1)$
- 添加删除元素需移动后面所有元素，平均 $O(n)$

字符串 String

链表 Linked List

- 单向链表
- 双向链表
- 循环链表

队列 Queue

- 优先级队列 Priority Queue
- 双向队列 Deque

栈 Stack

散列表 Hash Map

集合 Set

跳表 Skip List

二叉树 Binary Tree

- AVL 树
 - 左右子树高度差不大于 1
 - 为保持平衡频繁自旋
 - 磁盘 IO 多

- 平衡二叉查找树
 - 二叉查找树 Binary Search Tree
 - 所有节点是红色或黑色
 - 根节点为黑色
 - 叶子节点为黑色空节点
 - 红黑树 Red-black Tree
 - 每个红色节点的子节点都是黑色，叶子节点到根的路径上不能有连续红色节点
 - 任意节点到其每个叶子节点路径上黑色节点数量相同

树 Tree

完全二叉树

- 满二叉树
- 堆
 - 大顶堆 - 每个节点值大于等于其子树每个节点值
 - 小顶堆 - 每个节点值小于等于其子树每个节点值

平衡多路查找树

- Balance Tree
- B+ Tree
 - 中间节点只存储索引不存数据
 - 叶子节点有序
 - 遍历链表查找稳定
- B* Tree

拓扑排序 Topological Sorting

- 卡恩算法 Kahn
- DFS

图 Graph

- 深度优先搜索 Depth First Search
- 广度优先搜索 Breadth First Search

最短路径算法 Shortest Path Algorithm

- Dijkstra 算法
- Bellford 算法
- Floyd 算法

位图 Bit Map