

常见排序算法的特性，以及具体的处理过程

常见排序算法的时间复杂度和空间复杂度

常用的排序算法：（内部排序）所有排序流程都是在内存中完成

1、插入排序（直接插入排序和希尔排序）

直接插入排序，可用于链表结构

希尔排序，缩小增量排序，将数组分为多个按直接插入排序

2、选择排序（简单选择排序，堆排序）

简单选择排序：找到一个最小的数与第一个交换，然后将剩余的依次做同样操作

堆排序：根节点大于孩子节点为大顶，根节点小于孩子节点为小顶，如果大顶排序，则根节点为最大值，拿出该值，将剩余的在做大顶一次到最后一个即完成排序。排序以 $n/2$ 个节点开始调整，如果父节点小于子节点则兑换位置，然后 $n/2-1$ 个节点，一次完之后，将根节点和最后一个节点对掉，减掉最后一个即最大的一个，剩下的再做一次排序一次到最后一个

3、交换排序（冒泡排序、快速排序）

冒泡排序：第一个元素开始，一次比较，如果较小则交换，找出最小数，依次到最后一个元素

快速排序：二分查找

4、归并排序

使用合并操作完成排序

5、基数排序

先收集个位数填入0-9的表格，再按十位收集填入0-9表格，一直到所有元素最高位即完成排序

哈希表 (Hash) : 又称散列表, 杂凑表, 一种十分实用的查找技术, 具有极高的查找效率, 在进行存储和查找的时候需要用到哈希函数。什么叫做按内容存储。

哈希函数的构造方法, 没有特定要求, 只要是我们需要了解, 什么样的哈希函数才叫好的hash函数

直接地址法: 取某个关键之活关键字的某个线性函数值为哈希地址;

计数转换法: 将关键码看作是某个计数制上的整数, 然后将其转换为另一个基数制上的数

例: 对15, 30, 19进行基数转换法求哈希地址。

把三个数看成八进制转换为十进制是: 13, 24, 17

平方取中法

折叠法: 将关键码分成多段, 左边段向右折, 右边段向左折, 然后将他们叠加

移位法: 将关键码分为多段, 左边段右移, 右边段左移, 然后将他们叠加

随机数法

处理冲突的方法: 开放地址法 (线性探查法和双散列函数法)

线性探查法: 冲突发生后直接向下线性找一个新的空间存放

双散列函数法：当线性 $h = \text{key} \% p$ 得到值冲突后，则用 $h_i = (h + \text{key} \% (p-1)) \% p$ 记为二次散列

拉链法：将散列表的每个节点增加一个指针字段，用于链接同义词的子表，链表中的节点都是同义词

查找算法

顺序查找：指从表的一端开始，按顺序比对当前节点与关键字是否相等的一个查找方式；ASL平均查找长度是顺序查找的指标； $ASL = (n+1)/2$

二分查找（折半查找） $n/2$ 向上取整作为中间值，关键字与中间值对比，确定在左边找还是右边找；虽然效率高，但是适用范围小， $ASL = \log_2(n+1) - 1$

分块查找又称索引顺序查找：将表分为多段，每段按顺序，段内无序，索引表中存储该段的最大值和起始地址；查找先使用二分查找查找索引表，再使用顺序查找查找对应的段；二分查找索引表 $ASL = \log_2(n/s+1) + s/2$ ；段内顺序查找 $ASL = (s*s + 2s + n)/2s$