|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **排序类别** | **基本思想** | **排序算法** | **复杂度分析** | **稳定性** | **排序特点** |
| 插入排序 | 每次将一个待排序记录按其关键字大小插入到前面已排好的子序列中 | 直接插入排序 | 空间：O(1) | 稳定 | 适用于顺序与链式存储 |
|  |
| 折半插入排序 | 空间：O(1) | 稳定 | 仅仅减少了比较元素，比较次数与待排序表初始状态无关 |
|  |
| 希尔排序 | 时间复杂度依赖于增量序列的函数  O(n^1.3) | 不稳定，相等关键字记录被划分到不同的子表 | 确定增量：通过比较第一趟排序结果与初始条件，找第一个变续的关键字，再与该关键字原位置对比确定增量 |
| 交换排序 | 根据序列中两个元素关键字的比较结果来对换这两个记录在序列中的位置 | 冒泡排序 | 空间 O(1) | 稳定，相邻比较相等不换位 | 1. 用flag控制比较次数：若有flag，则比较次数与初始条件有关；若没有flag，则比较次数与初始条件无关 2. 冒泡排序中产生的有序子序列一定是全局有序的 |
|  |
| 快速排序（\*） | 空间：最坏情况下发生在两个区域分别包括n-1个元素和0个元素这种最大程度上的不对称发生在每层递归上 | 不稳定 | 1. 快排算法的性能主要取决与划分操作的好坏 2. 枢轴量的选择：第一个元素；头、尾、中间三个元素的中间值；随机选择 3. 内部排序算法中平均性能最优 4. 在快速排序中并不产生有序子序列，但每一趟都把一个元素放在最终位置上 |
|  |
| 选择排序 | 每一趟在选择一个特定元素放入特定位置 | 简单排序 | 空间 O(1) | 稳定 |  |
|  |
| 堆排序（\*） | 空间 O(1) | 不稳定 | 逻辑上的树形结构（完全二叉树），顺序存储从1开始 |
| 时间 O(n log2 n) |
| 归并排序 | 将两个或两个以上的有序表组合成一个有序表 | 2路归并（\*） | 空间 O(n) | 稳定 | M：趟数 K：路数 N：个数 |
| 时间 O(n log2 n) |
| 基数排序 | 多关键字排序，借助分配和收集两种操作 |  | 空间 O(r) | 稳定 | 按基数个数构造辅助队列（r），按关键字个数确定分配收集的操作次数 |
| 时间 O(d\*(n+r))  d:d趟收集分配；n:n个元素；r:r个队列 |