|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **算法** | **最好序列** | **最坏序列** | **平均时间** | **空间** | **稳定** |
| **交换** | 冒泡 | 有序O(n) | 逆序O(n2) | **O(n2)** | **O(1)** | ✔ |
| 快速(首元素为基准)  每趟找一个基准,low, high移动 | 基准元素为均匀划分 | 有序 | **O(nlog2n)** | **最好/最坏O(log2n)**  **O(n)** | **×** |
| **插入** | 直接插入 | 有序O(n) | 逆序O(n2) | **O(n2)** | **O(1)** | ✔ |
| 折半插入 | 比较次数减少  移动次数不变 | 比较次数减少  移动次数不变 | **O(n2)** | **O(1)** | ✔ |
| 希尔  给增量使其先部分有序再全局有序 | 顺序（取决于增量d） | 逆序（取决于增量d） | **O(n1.3~2)** | **O(1)** | **×** |
| **选择** | 简单选择  每一趟找最小的加入序列 | 有序  移动少,比较次数  (n-1)n / 2 | 逆序  移动多,比较次数  (n-1)n /2 | **O(n2)** | **O(1)** | **×** |
| 锦标赛 | / | / | **O(nlog2n)** | **O(n)** | **×** |
| 堆  建最大（增）或最小堆 | 建堆O(n)  排序O(nlog2n) | 建堆比较4n次  排序O(nlog2n) | **O(nlog2n)** | **O(1)** | **×** |
| **归并** | 两路归并 | 有序,每趟n  归并趟数log2n | 逆序，每趟n  归并趟数log2n | **O(nlog2n)** | **O(n)** | ✔ |
| 递归归并 | 有序,每趟n  归并趟数log2n | 逆序，每趟n  归并趟数log2n | **O(nlog2n)** | **O(n)**  递归为O(log2n) | ✔ |
| **基数** | 基数排序  (分配+收集) | - 关键字可拆分为d元组  - r较小  - **n较大** | n较小，d,r较大 | **O(d(n+r))**  d趟,每趟n次分配,r次收集 | **O(n+2r)**  n表示队列中开辟空间, r表示一个队列的两个指针 | ✔ |

**基于比较的，至少比较log2(n!)向上取整次**

**希尔排序：**

追求元素部分有序再逼近全局有序

d1 = n/2, d2 = d1/2.....

**快速排序：**

找一个元素作为基准，一趟后找到它的最终位置。Low和high移动，补空位。

时间复杂度：O(n\*递归深度), 空间复杂度：O(递归深度)

最深n, 最浅log2(n+1)向上取整

**基数排序：**

1. 不是基于比较的

排序算法选择：

1. 排序规模
2. 关键字分布
3. 稳定性要求

- 规模较小(<50)，直接插入或选择。前者比后者移动次数多，故选择优先

- 规模较大：

- 关键字分布随机：快速排序、堆（不会出现快排最坏情况）

- 接近有序：希尔（若不有序，移动次数多）

- 需要稳定：先用插入排序得到<50的有序段再归并。

- 基本有序：插入类或冒泡

- 规模很大但关键字位数少，基数排序



