

常用 sort 算法比较

排序方法	平均时间复杂度	最坏情况下时间复杂度	额外空间复杂度	稳定性
简单选择排序	$O(N^2)$	$O(N^2)$	$O(1)$	不稳定
冒泡排序	$O(N^2)$	$O(N^2)$	$O(1)$	稳定
直接插入排序	$O(N^2)$	$O(N^2)$	$O(1)$	稳定
希尔排序	$O(N^d)$	$O(N^2)$	$O(1)$	不稳定
堆排序	$O(N\log N)$	$O(N\log N)$	$O(1)$	不稳定
快速排序	$O(N\log N)$	$O(N^2)$	$O(\log N)$	不稳定
归并排序	$O(N\log N)$	$O(N\log N)$	$O(N)$	稳定
基数排序	$O(P(N+B))$	$O(P(N+B))$	$O(N+B)$	稳定

Search 和 Sort 是最基础的算法，基础但重要！

Search 的话，binary search 最简单实用。

Sort, 上面是常用 sort 对比，看图，其实也没啥可说的了。

注：1、bubble、selection 和 insert sort 统称为简单排序，优点是程序相对简单、好写，但时间复杂度较差 $O(N^2)$ ；

2、稳定排序（排序时，相同元素不调换位置）：bubble、insert、merge 和 radix sort 是稳定的；不稳定是因为元素调换位置时的间隔大于 1（选择排序和希尔排序可以较明显看出）；

3、Shell sort 基于 insert sort，其时间复杂度由其程序中的递减间隔序列决定，最好也是 $O(N^2)$ ；heap sort 基于 selection sort，看起来其时间复杂度还不错，但是实际中和快速排序哪个更快，还得看情况；

4、quick search, 可以称之为网红排序。数据量较大时可用, 但数据量较小时不如用插入排序。步骤: 选主元—划分子集—递归分治, 其中选主元可以用不同方法如: 直接取左|右端元素、Median3 等方法;

5、merge sort 除了要用额外空间 $O(N)$ 外, 都挺不错;

6、radix sort 基于 bucket sort, 只是按基数去选择 bucket, 在某些情况下, 其可以达到线性时间复杂度;

7、search 和 sort 除了在要用的时候灵活使用外, 重要的就是其时间、空间复杂度、稳定性、使用(情况)条件等。某些情况下(如面试), 这些算法(套路)及其特点(白纸上)写到、背到滚瓜烂熟, 会是一个漂亮的开始……

编程=算法+数据结构, 编程就像做菜, 数据结构是食材, 算法是食谱, 想把菜做好, 备上好食材、好食谱, 然后练吧。

面试时, 考官有可能让现场在黑板上写程序, 边写边和考官讨论效果较好。但前提是得准备好, 提前疯狂在白纸上和 PC 环境(leetcode 啥的)上刷题是最直接的方法。

恩……想必这些大家都知道, 可以, 就差行动了……

本渣要去学习, 88……