

剖析递归行为和递归行为时间复杂度的估算

一个递归行为的例子

master公式的使用

$$T(N) = a * T(N/b) + O(N^d)$$

- 1) $\log(b, a) > d \rightarrow$ 复杂度为 $O(N^{\log(b, a)})$
- 2) $\log(b, a) = d \rightarrow$ 复杂度为 $O(N^d * \log N)$
- 3) $\log(b, a) < d \rightarrow$ 复杂度为 $O(N^d)$

补充阅读: www.gocalf.com/blog/algorithm-complexity-and-master-theorem.html

堆排序的细节和复杂度分析

时间复杂度 $O(N \log N)$ ，额外空间复杂度 $O(1)$

堆结构非常重要

- 1, 堆结构的heapInsert与heapify
- 2, 堆结构的增大和减少
- 3, 如果只是建立堆的过程，时间复杂度为 $O(N)$
- 4, 优先级队列结构，就是堆结构

用数组表示堆
左孩子： $2*i+1$
右孩子： $2*i+2$
父亲： $(i-1)/2$

插入，删除
 $O(\log N)$

介绍一下工程中的综合排序算法

若你需要排序的是基本数据类型，则选择快速排序。

若你需要排序的是引用数据类型，则选择归并排序。

(基于稳定性考虑)

你需要排序的样本数量小于60，直接选择插入排序。

虽然插入排序的时间复杂度为 $O(N^2)$ ，我们是忽略常数项得出来的 $O(N^2)$ ，但在数量在60以内，插入排序的时间复杂度为 $O(N^2)$ 的劣势体现不出来，反而插入排序常数项很低，导致在小样本情况下，插入排序极快。 如果一开始数组容量很大，但可以分治处理，分治后如果数组容量 $(L > R - 60)$ 小于60，可以直接选择插排。