

# 递归和 $master$ 公式

前置知识：无

- 1) 从思想上理解递归：对于新手来说，递归去画调用图是非常重要的，有利于分析递归
- 2) 从实际上理解递归：递归不是玄学，底层是利用系统栈来实现的
- 3) 任何递归函数都一定可以改成非递归，不用系统帮你压栈（系统栈空间），自己压栈呗（内存空间）
- 4) 递归改成非递归的必要性：
  - a. 工程上几乎一定要改，除非确定数据量再大递归也一定不深，归并排序、快速排序、线段树、很多的平衡树等，后面都讲
  - b. 算法笔试或者比赛中（能通过就不改）
- 5)  $master$ 公式
  - a. 所有子问题规模相同的递归才能用 $master$ 公式， $T(n) = a * T(n/b) + O(n^c)$ ， $a$ 、 $b$ 、 $c$ 都是常数
  - b. 如果 $\log(b,a) < c$ ，复杂度为： $O(n^c)$
  - c. 如果 $\log(b,a) > c$ ，复杂度为： $O(n^{\log(b,a)})$
  - d. 如果 $\log(b,a) == c$ ，复杂度为： $O(n^c * \log n)$
- 6) 一个补充 $T(n) = 2 * T(n/2) + O(n * \log n)$ ，时间复杂度是 $O(n * ((\log n) \text{的平方}))$ ，证明过程比较复杂，记住即可