

# 归并分治

前置知识：讲解021-归并排序

原理：

- 1) 思考一个问题在大范围上的答案，是否等于，左部分的答案 + 右部分的答案 + 跨越左右产生的答案
- 2) 计算“跨越左右产生的答案”时，如果加上左、右各自有序这个设定，会不会获得计算的便利性
- 3) 如果以上两点都成立，那么该问题很可能被归并分治解决（话不说满，因为总有很毒的出题人）
- 4) 求解答案的过程中只需要加入归并排序的过程即可，因为要让左、右各自有序，来获得计算的便利性

补充：

- 1) 一些用归并分治解决的问题，往往也可以用线段树、树状数组等解法。时间复杂度也都是最优解，这些数据结构都会在【必备】或者【扩展】课程阶段讲到
- 2) 本节讲述的题目都是归并分治的常规题，难度不大。归并分治不仅可以解决简单问题，还可以解决很多较难的问题，只要符合上面说的特征。比如二维空间里任何两点间的最短距离问题，这个内容会在【挺难】课程阶段里讲述。顶级公司考这个问题的也很少，因为很难，但是这个问题本身并不冷门，来自《算法导论》原题
- 3) 还有一个常考的算法：“整块分治”。会在【必备】课程阶段讲到

聊：精妙又美丽的思想传统

# 归并分治-小和问题

假设数组  $s = [1, 3, 5, 2, 4, 6]$

在 $s[0]$ 的左边所有  $\leq s[0]$ 的数的总和为0

在 $s[1]$ 的左边所有  $\leq s[1]$ 的数的总和为1

在 $s[2]$ 的左边所有  $\leq s[2]$ 的数的总和为4

在 $s[3]$ 的左边所有  $\leq s[3]$ 的数的总和为1

在 $s[4]$ 的左边所有  $\leq s[4]$ 的数的总和为6

在 $s[5]$ 的左边所有  $\leq s[5]$ 的数的总和为15

所以 $s$ 数组的“小和”为： $0 + 1 + 4 + 1 + 6 + 15 = 27$

给定一个数组 $arr$ ，实现函数返回 $arr$ 的“小和”

测试链接：

<https://www.nowcoder.com/practice/edfe05a1d45c4ea89101d936cac32469>

课上解法和归并排序的时间复杂度一样 $O(n * \log n)$

# 归并分治-翻转对数量

给定一个数组 `nums` ,

如果  $i < j$  且  $nums[i] > 2 * nums[j]$  我们就将  $(i, j)$  称作一个翻转对

你需要返回给定数组中的翻转对的数量

测试链接：

<https://leetcode.cn/problems/reverse-pairs/>

课上解法和归并排序的时间复杂度一样  $O(n * \log n)$