

# 分治 (Divide And Conquer)

■ 分治，也就是分而治之。它的一般步骤是

- ① 将原问题分解成若干个规模较小的子问题（子问题和原问题的结构一样，只是规模不一样）
- ② 子问题又不断分解成规模更小的子问题，直到不能再分解（直到可以轻易计算出子问题的解）
- ③ 利用子问题的解推导出原问题的解

■ 因此，分治策略非常适合用递归

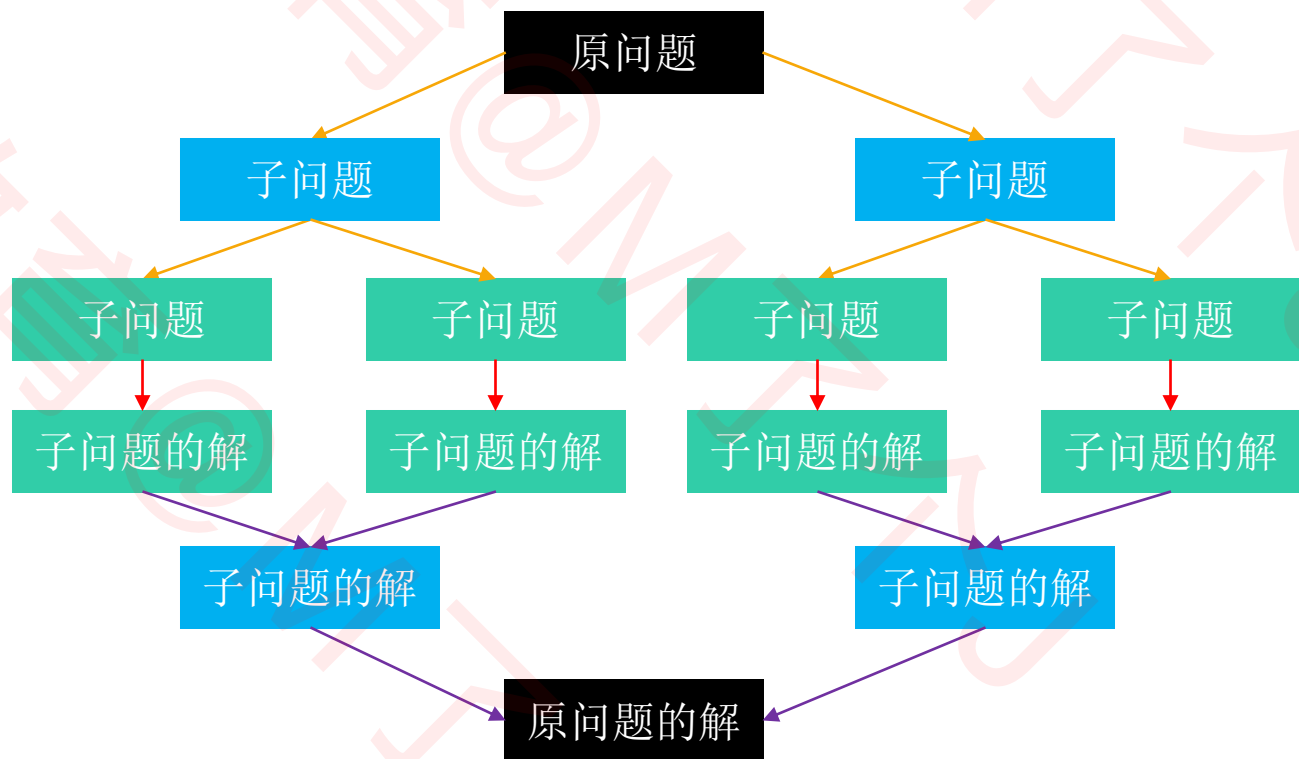
■ 需要注意的是：子问题之间是相互独立的

■ 分治的应用

□ 快速排序

□ 归并排序

□ Karatsuba算法（大数乘法）



# 主定理 (Master Theorem)

■ 分治策略通常遵守一种通用模式

□ 解决规模为  $n$  的问题，分解成  $a$  个规模为  $\frac{n}{b}$  的子问题，然后在  $O(n^d)$  时间内将子问题的解合并起来

□ 算法运行时间为： $T(n) = aT\left(\frac{n}{b}\right) + O(n^d)$ ,  $a > 0, b > 1, d \geq 0$

✓  $d > \log_b a$ ,  $T(n) = O(n^d)$

✓  $d = \log_b a$ ,  $T(n) = O(n^d \log n)$

✓  $d < \log_b a$ ,  $T(n) = O(n^{\log_b a})$

■ 比如归并排序的运行时间是： $T(n) = 2T\left(\frac{n}{2}\right) + O(n)$ ,  $a = 2, b = 2, d = 1$ , 所以  $T(n) = O(n \log n)$

■ 思考：为什么有些问题采取分治策略后，性能会有所提升？