用 Karatsuba 算法计算多项式乘法

给定两个多项式

$$
\begin{aligned}
A(x) &= \sum_{k=0}^{n-1} a_k x^k = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^{n-1}, \\
B(x) &= \sum_{k=0}^{m-1} a_k x^k = b_0 + b_1 x + b_2 x^2 + \cdots + b_m x^{m-1},
\end{aligned}
$$

不妨假设 $n \geqslant m$。如果 $n < m$ 就交换 $A(x)$ 和 $B(x)$。

将 $A(x)$ 分成两部分：

$$A(x) = A_l(x) + x^{\lfloor n/2 \rfloor} A_h(x)$$

同样地，将 $B(x)$ 分成两部分：

$$B(x) = B_l(x) + x^{\lfloor n/2 \rfloor} B_h(x)$$

计算以下 3 个乘积：

$$
\begin{aligned}
z_0(x) &= A_l(x) B_l(x), \\
z_1(x) &= (A_l(x) + A_h(x))(B_l(x) + B_h(x)), \\
z_2(x) &= A_h(x) B_h(x).
\end{aligned}
$$

最后的乘积为：

$$A(x)B(x) = z_0(x) + (z_1(x) - z_0(x) - z_2(x)) x^{\lfloor n/2 \rfloor} + z_2(x) x^{2 \cdot \lfloor n/2 \rfloor}.$$

所以我们可以将一个规模为 n 的问题分解为三个规模为 $n/2$ 的问题，在 $\Theta(n)$ 的时间里分解、合并，因此该算法的运行时间为

$$T(n) = 3T(n/2) + \Theta(n) = \Theta\left(n^{\log_2 3}\right).$$

注意，这个算法很慢（和 FFT 相比），不加任何优化的情况下无法通过后十组测试数据。相比之下，$O(nm)$ 的暴力算法虽然时间复杂度更高，但它的实际运行时间具有较小的常数因子，在 n 和 m 比较小的时候表现非常优秀。你能否据此设计出一种 Karatsuba 的优化？