# Karatsuba O(n^1.585)

Directly: n digits, $\Theta(n^2)$ runtime

Bad algorithm: a+a+ ... + a (b times), $O(n \cdot 10^n)$

Let's use divide and conquer.

$a = a_h \cdot 10^{\frac{n}{2}} + a_l$

$b = b_h \cdot 10^{\frac{n}{2}} + b_l$

$a \cdot b = a_h \cdot b_h \cdot 10^n + (a_h \cdot b_l + a_l \cdot b_h) \cdot 10^{\frac{n}{2}} + a_l \cdot b_l$

First attempt:

MULT(a, b):

1. Recursively compute: MULT($a_h, b_h$), MULT($a_h, b_l$), MULT($a_l, b_h$), MULT($a_l, b_l$)
2. Output: Runtime $T(n) = 4 \cdot T(\frac{n}{2}) + \Theta(n)$
   $n + 2n + 4n + ...n^2 = \Theta(n^2)$

Actual idea: $a_h \cdot b_l + a_l \cdot b_h = M - H - L$

1. $H = a_h \cdot b_h$
2. $M = (a_h + a_l) \cdot (b_h + b_l)$
3. $L = a_l \cdot b_l$

MULT(a, b):

1. Recursively compute: MULT($a_h, b_h$), MULT($a_h + a_l, b_h + b_l$), MULT($a_l, b_l$)
2. Output: Runtime $T(n) = 3 \cdot T(\frac{n}{2}) + \Theta(n)$
   $n + \frac{3}{2}n + \frac{9}{4}n + ...(\frac{3}{2})^{log_2^n} n = \Theta(n^{1.585})$

Followup work:

- Toom-Cook (1963): $T(n) = 5 \cdot T(\frac{n}{3}) + \Theta(n) = \Theta(n^{1.465})$
- Schomhage-Strassen (1971): $\Theta(nlogn \cdot loglogn)$
- Furer (2007): $\Theta(nlogn \cdot 2^{log_n^*})$, $log_n^*$: recursively take log of n until 1
- Harvey-de Hoeven (2019): $\Theta(nlogn)$