What is...Karatsuba's algorithm?

Or: Faster than expected

Classical multiplication

(x - 3)(4x – 5)

	x	-3
4x	$4x^2$	-12x
-5	-5x	15

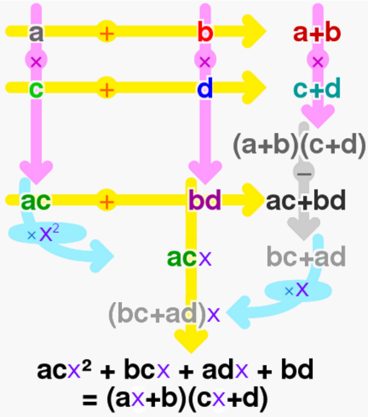
$4x^2$ - 12x – 5x + 15

$4x^2$ - 17x + 15

	x^2	$-4x$	-2
$2x^2$	$2x^4$	$-8x^3$	$-4x^2$
$-x$	$-x^3$	$4x^2$	$2x$
-1	$-x^2$	$4x$	2

- Given two polynomials f and g of degree $< n$; we want fg
- Classical polynomial multiplication needs n^2 multiplications and $(n-1)^2$ additions; thus $mult(poly) \in O(n^2)$
- It doesn't appear that we can do faster

Using more operations...



- **Karatsuba ∼1960** It gets faster!
- We compute ac, bd, $u = (a+b)(c+d)$, $v = ac + bd$, $u - v$ with 3 multiplications and 4 additions = 7 operations – more than before

  Upshot We only have 3 multiplications not 4

...but fewer multiplications

ALGORITHM 8.1 Karatsuba's polynomial multiplication algorithm.

Input: $f, g \in R[x]$ of degrees less than n, where R is a ring (commutative, with 1) and n a power of 2.

Output: $fg \in R[x]$.

1. **if** $n = 1$ **then return** $f \cdot g \in R$
2. let $f = F_1 x^{n/2} + F_0$ and $g = G_1 x^{n/2} + G_0$, with $F_0, F_1, G_0, G_1 \in R[x]$ of degrees less than $n/2$
3. compute $F_0 G_0$, $F_1 G_1$, and $(F_0 + F_1)(G_0 + G_1)$ by a recursive call
4. **return** $F_1 G_1 x^n + ((F_0 + F_1)(G_0 + G_1) - F_0 G_0 - F_1 G_1) x^{n/2} + F_0 G_0$

Example

$f = g = x^3 + x^2 + x + 1$ is equal to $F_1 + F_0 = (x+1)x^2 + x + 1$

$F_0^2 = F_1^2 = (x+1)^2$ and $(2x+2)(2x+2)$ need 7 ops = 21 ops

To get fg we then need two more ops = 23 ops

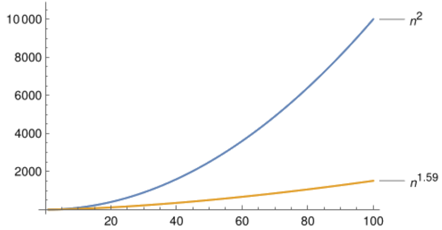
Classical we need $4^2 + (4-1)^2 = 25$ ops

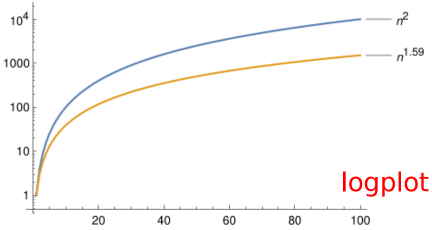
Enter, the theorem

Karatsuba ∼1960 Using k-adic expansion, this works for numbers as well

Theorem (Karatsuba ∼1960) For $n = 2^k$ (n=#digits) we have $mult \in O(n^{1.59})$

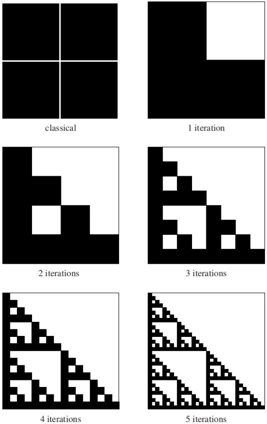
Ditto for polymult





- Multiplication is everywhere so this is fabulous
- There is also a version for general n but the analysis is somewhat more involved
- Nowadays computer algebra systems have (beefed-up versions of) Karatsuba's algorithm build in

A picture why this is faster



8.2: Cost (= black area) of Karatsuba's algorithm for increasing recursion depths.
ige approaches a fractal of dimension $\log_2 3 \approx 1.59$.

- The above (ignoring additions) shows why this is much faster
- For $n = 2^k$ we have $mult(poly) \in O(n^{1.59})$ $(1.59 \approx \log(3))$

Thank you for your attention!

I hope that was of some help.