What is...discrete analysis?

Or: Subfields of mathematics 31

Integer multiplication

(x - 3)(4x – 5)

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

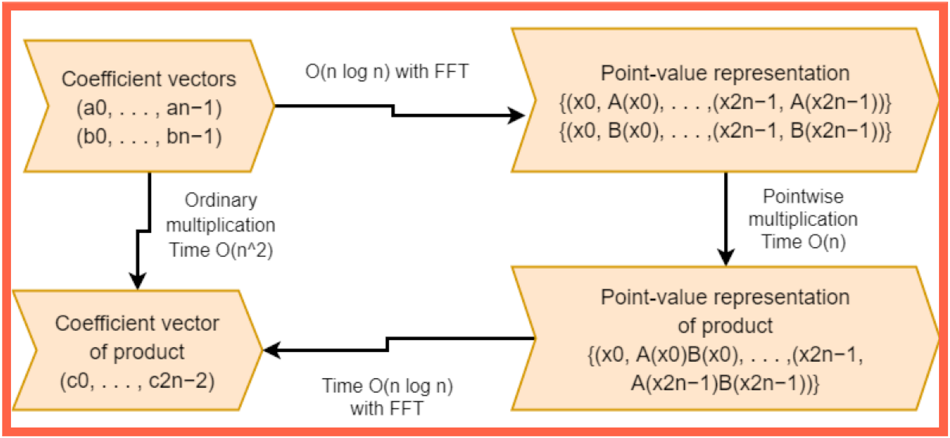
$4x^2$ - 12x – 5x + 15

$4x^2$ - 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
- Naive polynomial multiplication needs n^2 multiplications and $(n-1)^2$ additions; thus $mult(poly) \in O(n^2)$
- Ditto for integer multiplication with $n =$ number or digits

Fast multiplication?

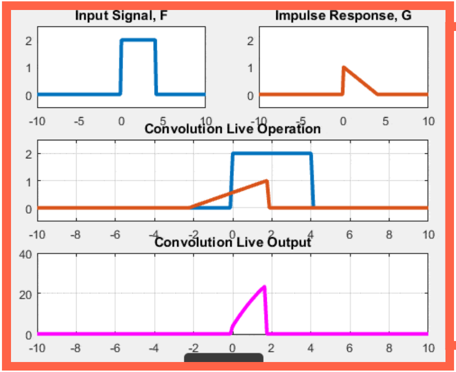


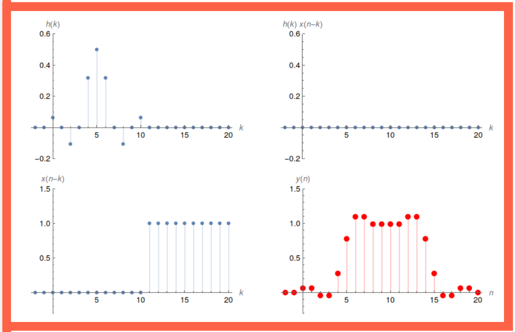
- Assume that there is an operation DFT_ω such that:

$$fg = DFT_\omega^{-1}\big(DFT_\omega(f)DFT_\omega(g)\big)$$

  with DFT_ω and DFT_ω^{-1} and $DFT_\omega(f)DFT_\omega(g)$ being cheap
- Then compute fg for polynomials f and g is cheap

DFT = discrete FT (= Fourier transform)





- Fourier transform translates between convolution (“pointwise”) and multiplication of functions
- Slogan Convolution = area obtained by sliding f through g
- Essentially, the DFT (group $\mathbb{Z}/n\mathbb{Z}$) is a discrete version of the FT (group $\mathbb{Z}$)

Enter, the theorem

FFT = fast FT computes the DFT and its inverse in $O(n \log n)$

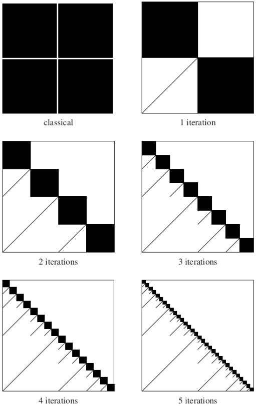
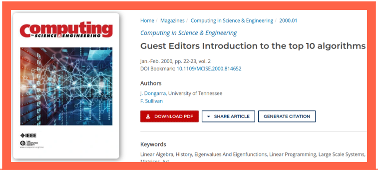


FIGURE 8.5: Cost of the FFT for increasing recursion depths. The black area is proportional to the total work.

- $\Rightarrow$ Integer multiplication is in $O(n \log n)$
- Discrete analysis answers similar questions!

Algorithms of the century



- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method

- Above From the IEEE Computer Society Journal
- No such list can be perfect but that FFT/DFT made it on it should tell us something ☺

Thank you for your attention!

I hope that was of some help.