Polynommultiplikation

Polynome (vom Grad $n - 1$) kann man auf unterschiedliche Weise beschreiben:

- Entweder durch Angabe von n Koeffizienten $a_0, \ldots, a_{n-1}$ mit $p(x) = a_{n-1}x_{n-1} + \ldots + a_1 x + a_0$.
- Oder durch Angabe von n verschiedenen x-Werten und den zugehörigen Werten des Polynoms, also n Paaren $(x_1, p(x_1)), \ldots, (x_n, p(x_n))$.

Idee: In der *zweiten* Darstellung ist die Berechnung des Produktes zweier Polynome viel einfacher!!

Polynommultiplikation: Schematische Darstellung

$$
\begin{tikzcd}
\begin{array}{c} \vec{a} = (a_0, a_1, \ldots, a_{n-1}) \\ \vec{b} = (b_0, b_1, \ldots, b_{n-1}) \end{array} \arrow[r, dotted, "\text{1.Ansatz}"] \arrow[d, "\text{DFT}"'] & \vec{c} = IDFT(\vec{\psi}, \omega) \\
\begin{array}{c} \vec{\xi} = DFT(\vec{a}, \omega) \\ \vec{\phi} = DFT(\vec{b}, \omega) \end{array} \arrow[r, "\text{komponentenweise}", "\text{Multiplikation}"'] & \vec{\psi} = (\xi_0 \cdot \phi_0, \ldots, \xi_{n-1} \cdot \phi_{n-1}) \arrow[u, "\text{IDFT}"']
\end{tikzcd}
$$

Polynommultiplikation mit Hilfe der diskreten Fouriertransformierten

Für einen Vektor $a = (a_0, \ldots, a_{n-1})$ bezeichne $p_a(x)$ das Polynom

$$p_a(x) = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1}.$$

Die *diskrete Fouriertransformierte* von $\vec{a}$ an der Stelle η bezeichnet den Vektor

$$DFT(\vec{a}, \eta) = (p_{\vec{a}}(1), p_{\vec{a}}(\eta), \ldots, p_{\vec{a}}(\eta^{n-1})).$$

Umgekehrt ist für einen Vektor $\vec{\xi} = (\xi_0, \ldots, \xi_{n-1})$ die *inverse diskrete Fouriertransformierte* definiert durch

$$IDFT(\vec{\xi}, \eta) = \vec{a}, \quad \text{so dass } p_{\vec{a}}(\eta^i) = \xi_i \text{ für alle } i = 0, \ldots, n-1.$$

Rechnen mit großen Zahlen

Eingabe: a, b n-Bit Zahlen

Addition:

$O(n)$ „normale" Additionen (Maschinenbefehle)

Multiplikation:

Schulmethode: $O(n^2)$ „normale" Additionen und Multiplikationen

Karatsuba: $O(n^{1.59})$ „normale" Additionen und Multiplikationen

Schönhage, Strassen: $O(n \log n \log \log n)$ „normale" Additionen und Multiplikationen
(verwendet die schnelle Polynommultiplikation mit DFT)

Division:

Man kann zeigen: $T_{Division}(n) = O(T_{Multiplikation}(n))$

Potenzen mod n: a^x mod n

abwechselndes Multiplizieren und Quadrieren: $O(n)$ Multiplikationen und
Modulo-Berechnungen von n-Bit Zahlen

Optimierungsprobleme

Ein Optimierungsproblem Π kann man als ein 4-Tuple $\langle \mathcal{I}, Sol, val, \text{goal}\rangle$ beschreiben:

- $\mathcal{I} \subseteq \Sigma^*$ ist die Menge der *Instanzen*; wobei wir annehmen, dass man die Frage „Ist $x \in \mathcal{I}$?“ in polynomieller Zeit entscheiden kann.
- Zu jeder Instanz $I \in \mathcal{I}$ bezeichnet $Sol(I) \subseteq \Sigma^*$ die (nichtleere) Menge der zulässigen *Lösungen*.
- Zu jeder Instance I und Lösung $x \in Sol(I)$ ist $val(I, x)$ eine positive ganze Zahl. Die Funktion $val(\cdot, \cdot)$ nennt man *Zielfunktion*.
- goal $\in \{\min, \max\}$.