Grundzüge von Datenstrukturen und Algorithmen (WS 2013/2014)

Lösungsvorschlag zu Aufgabenblatt 8

Aufgabe 1

Wir wollen die Kosten $K(n)$ für $F_S(x) = \prod_{s \in S}(x - s)$ berechnen. Für $n := |S| = 1$ ist offensichtlich:

$$K(n) \in \mathcal{O}(1) \tag{1}$$

Wir betrachten nun den Fall $n > 1$. Wir partitionieren S in S' und S'', so dass gilt:

$$S = S' \cup S'' \wedge S' \cap S'' = \varnothing \wedge |S'| = \lceil \frac{n}{2} \rceil \tag{2}$$

Offensichtlich gilt $F_S(x) = F_{S'} \cdot F_{S''}$
Wie aus Aufgabenstellung kann das Produktpolynom zweier Polynome A und B vom Grad n in $\mathcal{O}(n \cdot log(n))$ berechnet werden. Sowohl $F_{S'}(x)$ als auch $F_{S''}(x)$ sind offensichtlich vom Grad $\leq \lceil \frac{n}{2} \rceil$. Damit lässt sich ihr Produkt in $\mathcal{O}(\frac{n}{2} \cdot log(\frac{n}{2}))$ berechnen.
Damit gilt für $K(n), n > 1$:

$$\begin{aligned}
K(n) &= K(\lceil \frac{n}{2} \rceil) + K(\lfloor \frac{n}{2} \rfloor) + \mathcal{O}(\frac{n}{2} \cdot \log(\frac{n}{2})) && (3)\\
&\leq 2 \cdot K(\lceil \frac{n}{2} \rceil) + \mathcal{O}(\frac{n}{2} \cdot \log(\frac{n}{2})) && (4)\\
&\leq 2 \cdot K(\lceil \frac{n}{2} \rceil) + \alpha \cdot (\frac{n}{2} \cdot \log(\frac{n}{2})) \qquad | \text{ für hinreichend großes } \alpha && (5)\\
&\leq 2 \cdot K(\lceil \frac{n}{2} \rceil) + \alpha \cdot \frac{n}{2} \cdot \log n && (6)
\end{aligned}$$

Aus 1 und 6 folgt, dass man den 2. Fall des Mastertheorems mit $a = 2, b = 2, \beta = \frac{\alpha}{2}, c = 1, k = 1$ anwenden kann (da $a = 2 = b^c$). Dann gilt:

$$\begin{aligned}
K(n) &\in \mathcal{O}(n^c \cdot log^{k+1}(n) && (7)\\
&= \mathcal{O}(n \cdot \log^2(n)) && (8)
\end{aligned}$$

Aufgabe 2

Fallunterscheidung: Für $n = 1$ ist $F_U = v_1$ (Grad 0)

Für $n > 1$ werden die Mengen U' und U'' gebildet, wie in der Aufgabenstellung beschrieben. Nun soll $F_{U'}$ berechnet werden:

1. Bestimme Polynom N (mit Grad $\lceil \frac{n}{2} \rceil$), welches (genau) alle geforderten Nullstellen ergibt. Laufzeit: $\mathcal{O}(\lceil \frac{n}{2} \log^2 \frac{n}{2} \rceil) = \mathcal{O}(n \log^2 n)$.

2. Berechne für jedes $(s_i, v_i) \in U'$, bei dem v_i nicht auf 0 gesetzt wurde, $n_i = N(s_i)$. Dies kann durch die gegebene Simultanevaluation in $\mathcal{O}(n \log^2 n)$ erledigt werden.

3. Wir wollen erreichen, $F_{U'} = N \cdot \bar{N}$. Für die Nullstellen ist durch N garantiert; durch den Grad von N und die Angabe, dass alle s_i paarweise verschieden sind, kann N keine Nullstellen an Stellen haben, die nach U' als nicht-null gefordert ist.

   $\bar{N}$ soll also durch die Punkte $\bar{N}_{Pkt} = \left\{\left(s_i, \frac{v_i}{n_i}\right) \mid (s_i, v_i) \in U' \land v_i \neq 0\right\}$ verlaufen. Die Berechnung dieser Menge bzw. Werte durch einfache Iteration über U' braucht $\mathcal{O}(n)$ Zeit.

4. Rekursion auf $\bar{N}_{Pkt}$, um $\bar{N}$ zu bestimmen: $T\left(\frac{n}{2}\right)$ Zeit.

5. Berechne $N \cdot \bar{N}$, um $F_{U'}$ zu erhalten: Laufzeit als $\mathcal{O}(n \log n)$ gegeben.

6. Analoges Vorgehen, um $F_{U''}$ zu bestimmen

7. Berechne $F_{U'} + F_{U''}$, um F_U zu erhalten: $\mathcal{O}(n)$, da fast gleicher Grad (je nach Rundung von $\frac{n}{2}$).

Die Korrektheit folgt aus der Korrektheit der Schritte, und den kleiner werdenden Schritten (die Rekursion terminiert).

Aus dem Algorithmus ist eine Rekurrenz für die Laufzeit ablesbar:

$$T(n) \leq \begin{cases} \mathcal{O}(1) & n = 1 \\ \underbrace{\mathcal{O}(n)}_{\text{Split}} + \left(\underbrace{2 \cdot \mathcal{O}(n \log^2 n)}_{\text{1.,2.}} + \underbrace{\mathcal{O}(n)}_{\text{3.}} + \underbrace{T\left(\frac{n}{2}\right)}_{\text{4.}} + \underbrace{\mathcal{O}(n \log n)}_{\text{5.}} \right) \underbrace{\cdot 2}_{\text{6.}} + \underbrace{\mathcal{O}(n)}_{\text{7.}} & n > 1 \end{cases}$$

$$= \begin{cases} \mathcal{O}(1) & n = 1 \\ 2T\left(\frac{n}{2}\right) + \mathcal{O}(n \log^2 n) & n > 1 \end{cases}$$

Daraus lässt sich über Master's Theorem schließen: $T \in \mathcal{O}(n \log^3 n)$

Grundzüge DS & Alg (WS14/15)

Lösungsvorschlag zu Aufgabenblatt 9

Aufgabe 3

Zu zeigen: $P(x) \cdot B(x)$ und $P(x) \bmod B(x)$ lassen sich in $O(m)$ berechnen.

Bei $P(x) \cdot B(x) = P(x) \cdot (x^l - a)$:
Initialisiere ein neues Koeffizientenarray B der Größe $m + l$ mit $\forall 0 \leq i < m + l : B[i] = 0$.
Wir nehmen an dies geschieht in $O(1)$.
Nun durchlaufen wir das Koeffizientenarray A von $P(x)$ und setzen $B[l + i] = A[i]$.
Damit enthält B die Koeffizienten des Polynoms $P(x) \cdot x^l$ in $O(m)$.
Folglich muss nun noch $P(x) \cdot a$ subtrahiert werden. Dafür durchlaufen wir A und setzen $B[i] = B[i] - A[i] \cdot a$.
Somit erhalten wir das Koeffizientenarray von $P(x) \cdot B(x)$ in $O(m)$.

Bei $P(x) \bmod B(x) = P(x) \bmod (x^l - a)$:
$P(x)$ ist vom Grad m und A sein Koeffizientenarray.
Nun subtrahieren wir von diesem Polynom
$a_m \cdot x^{m-l} \cdot (x^l - a) = a_m \cdot x^m - a \cdot a_m \cdot x^{m-l}$ folgendermaßen:
Setze $A[m - l] = A[m - l] + a \cdot A[m]$ und $A[m] = 0$
Dies alles geschieht in $O(1)$ und wir erhalten ein Polynom mit Grad $m - 1$.
Dies setzt man nun iterativ fort, bis man ein Polynom des Grades $l - 1$ erhält. Dieses stellt das Polynom $P(x) \bmod x^l - a$ dar. Im worst-case ist $l = 1$, man benötigt dann m konstante Iterationen und hat eine Gesamtlaufzeit von $O(m)$.

Aufgabe 4

Wir definieren $m = n - 1$.

Sei $p(x) = p_0 + p_1 \cdot x + ... + p_{n-1} \cdot x^{n-1}$, $S_n = \{\omega_n^i | 0 \leq i < n\}$ und $n' = n/2$.
Dann ist

$$S_{n'} = \{(\omega_n^2)^i | 0 \leq i < n'\}$$

Weil:

$$(\omega_n^{2i})^{n'} = (\omega_n^n)^i = 1$$

Seien nun außerdem:

$$p_e(z) = p_0 + p_2 \cdot z + \cdots + p_{n-2} \cdot z^{n'-1}$$

$$p_o(z) = p_1 + p_3 \cdot z + \cdots + p_{n-1} \cdot z^{n'-1}$$

Dann gilt:

$$p(x) = p_e(x^2) + x \cdot p_o(x^2)$$

Wir benutzen nun divide-and-conquer, indem wir $S_n \rightarrow S_{n'}$ und $p \rightarrow p_e, p_o$ verwenden.

Wir suchen

$$p(S_n) = [p(\omega_n^0), p(\omega_n^1), \ldots, p(\omega_n^{n-1})] = \bar{p}$$

Dazu berechnen wir rekursiv:

$$\bar{u} = p_e(S_{n'})$$
$$\bar{v} = p_o(S_{n'})$$

Es gilt für alle $0 \leq i < n'$:
$\bar{p}_i = p(\omega_n^i) = p_e(\omega_n^{2i}) + \omega_n^i \cdot p_o(\omega_n^{2i}) = \bar{u}_i + \omega_n^i \cdot \bar{v}_i$

$\bar{p}_{i+n'} = p(\omega_n^{i+n'}) = p_e(\omega_n^{2i+2n'}) + \omega_n^i \cdot \underbrace{\omega_n^{n'}}_{=-1} \cdot p_o(\omega_n^{2i+2n'}) =$
$p_e(\omega_n^{2i+2n'}) - \omega_n^i \cdot p_o(\omega_n^{2i+2n'}) = \bar{u}_i - \omega_n^i \cdot \bar{v}_i$

Folglich gilt:

$$\bar{p} = [\bar{u} + \omega_n \cdot \bar{v}, \bar{u} - \omega_n \cdot \bar{v}]$$

Lauzeit: Um aus $\bar{u}$ und $\bar{v}$ $\bar{p}$ berechnen zu können brauchen wir $O(n)$.

Daraus ergibt sich für die Laufzeit: $T(n) \leq O(n) + 2 \cdot T(n/2)$
$\Longrightarrow T(n) \in O(n \cdot log(n))$

Aufgabe 5

Wir lösen das allgemeinere Problem:

$$P(x) = x^n - w^l$$

mit $n = 2^k$, $k \in \mathbb{Z}$, w ist eine beliebige Einheitswurzel von Grad d, $d \in \mathbb{Z}$ und $s_1, s_2, ...s_n$ sind Nullstellen von $P(x)$. Also: $P(x) = (x - s_1)(x - s_2)...(x - s_n)$. Wir möchten nun das folgene Interpolationsproblem lösen und P_U finden:

$$U = \{(s_j, v_j)|1 \leq j \leq n\}$$

Mit $w = 1$ erhalten wir unser ursprüngliches Problem. Wir gehen ähnlich zur Aufgabe 2 vor, erhalten aber durch die zusätzlichen Einschränkungen von s_j eine bessere Laufzeit.

Zunächst teilen wir unser gesuchtes Polynom $P(x)$ wie folgt auf:

$$P(x) = x^n - w^l = x^{2m} - w'^{2l} = (x^m - w'^l)(x^m - w'^{l+d})$$

w' ist eine der beiden Wurzeln von w: $w = w'^2$. Damit ist w' eine Einheitswurzel vom Grad $2d$, $w'^{2d} = 1, w'^d = -1$ und $m = n/2$. Wir setzen oBdA

$$P_1(x) = x^m - w'^l = (x - s_1)(x - s_2)...(x - s_m)$$

$$P_2(x) = x^m - w'^{l+d} = (x - s_{m+1})(x - s_{m+2})...(x - s_n)$$

Damit haben wir $P(x)$ in zwei Polynome von Grad $n/2$ geteilt. $P(x) = P_1(x)P_2(x)$.

So wie in Aufgabe 2 wird U in U' und U'' geteilt.

$$U' = \{(s_j, 0)|1 \leq j \leq m\} \cup \{(s_j, v_j)|m < j \leq n\}$$

$$U' = \{(s_j, v_j)|1 \leq j \leq m\} \cup \{(s_j, 0)|m < j \leq n\}$$

Es gilt weiterhin (wie in Aufgabe 2) $F_U = F_{U'} + F_{U''}$ (O(1)). Im Folgenden wird nur die Berechnung von $F_{U'}$ betrachtet ($F_{U''}$ analog). $F_{U'}$ kann dargestellt werden als

$$F_{U'}(x) = P_1(x)N_1(x) \qquad O(n)\ da\ P_1(x)\ binomial$$

Für $s_j, 1 \leq j \leq m$ gilt diese Gleichung bereits, ohne dass N_1 bekannt ist, da $P(s_j) = (x - s_1)(x - s_2)...(x - s_m) = 0$ für $x \in \{s_1, s_2...s_m\}$. Aber wir müssen noch dafür sorgen, dass $F_{U'}(x) = P_1(x)N_1(x)$ auch für $x \in \{s_{m+1}, s_{m+2}...s_n\}$ gilt. Es muss also $N_1(x)$ abhängig von $P_1(x)$ gewählt werden.

$$P_1(s_j) = s_j{}^m - w'^l, m < j \leq n$$

Mit dem Wissen das $P_2(s_j) = s_j{}^m - w'^{l+d} = 0$ für $m < j \leq n$ folgt $s_j{}^m = w'^{l+d}$ und damit

$$P_1(s_j) = w'^{l+d} - w'^l, m < j \leq n, w'^{l+d} - w'^l const. \qquad O(1)$$

N_1 ist damit das Polynom der Interpolation der Menge $\{(s_{m+1}, \frac{v_{m+1}}{P_1(s_{m+1})}), (s_{m+2}, \frac{v_{m+2}}{P_1(s_{m+2})})...(s_n, \frac{v_n}{P_1(s_n)})\} = \{(s_{m+1}, \frac{v_{m+1}}{w'^l}), (s_{m+2}, \frac{v_{m+2}}{w'^l})...(s_n, \frac{v_n}{w'^l})\}$. Dies ist ähnlich zu der Ausgangsmenge U und kann damit in $T(\frac{n}{2})$ berechnet werden.

Gesamte Laufzeit: $T(n) = O(n) + 2T(\frac{n}{2}) \Rightarrow T \in O(n \log n)$

Aufgabe 6

Aufgabe 4 beschreibt die Evaluation eines Polynoms. Dabei wird das Polynom von der Koeffizientendarstellung $A(x) = a_0 + a_1x^1 + a_2x^2 + ... + a_nx^n$ in die Point-Value-Darstellung $\{(x_0, y_0), (x_1, y_1), ..., (x_{n-1}, y_{n-1})\}$ gebracht. Dies ist in Zeit $O(n \log n)$ möglich.
Die Multiplikation in Point-Value-Darstellung läuft in $O(n)$:
$\{(x_0, y_0), ..., (x_{n-1}, y_n - 1)\} \cdot \{(x_0, z_0), ..., (x_{n-1}, z_{n-1})\} = \{(x_0, y_0 \cdot z_0), ..., (x_{n-1}, y_{n-1} \cdot z_{n-1})\}$
Die Interpolation zurück zur Koeffizientendarstellung dauert, wie in Aufgabe 5 gezeigt, $O(n \log n)$.

Somit kommt man auf eine Gesamtlaufzeit von $2O(n \log n) + O(n) \Rightarrow O(n \log n)$.

Die Multiplikation direkt in der Koeffizientendarstellung hat dagegen mit einer naiven Implementierung eine Laufzeit von $O(n^2)$.