

Complexiteit

Uitwerkingen opgave 47 + 48c

opgave 47

a. We moeten het polynoom $p(x) = x^7 + 4x^6 - 8x^4 + 6x^3 + 9x^2 + 2x - 3$ in de juiste vorm brengen. We zoeken dus naar polynomen q en r zodanig dat $p(x) = (x^4 + b)q(x) + r(x)$, waarbij q en r beide monische derdegraadspolynomen zijn. Als we de coëfficiënt van x^i in q als q_i noteren (en evenzo voor $p(x)$ en $r(x)$), krijgen we de vergelijking:

$$\begin{aligned} p(x) &= (x^4 + b)(x^3 + q_2x^2 + q_1x + q_0) + x^3 + r_2x^2 + r_1x + r_0 \\ &= x^7 + q_2x^6 + q_1x^5 + q_0x^4 + (1+b)x^3 + (r_2 + bq_2)x^2 + (r_1 + bq_1)x + bq_0 + r_0 \end{aligned}$$

Voor iedere coëfficiënt q_i en r_i kunnen we nu de waarde vinden, door de overeenkomstige coëfficiënten af te lezen uit $p(x) = x^7 + 4x^6 - 8x^4 + 6x^3 + 9x^2 + 2x - 3$ en de vergelijkingen op te lossen. We constateren eerst dat $1 + b = p_3$, dus dat $b = p_3 - 1 = 6 - 1 = 5$ en vervolgens:

$$\begin{array}{ll} q_0 = p_4 = -8 & r_0 = p_0 - bq_0 = -3 - 5 \cdot -8 = 37 \\ q_1 = p_5 = 0 & r_1 = p_1 - bq_1 = 2 - 5 \cdot 0 = 2 \\ q_2 = p_6 = 4 & r_2 = p_2 - bq_2 = 9 - 5 \cdot 4 = -11 \end{array}$$

Samen krijgen we $q(x) = x^3 + 4x^2 - 8$ en $r(x) = x^3 - 11x^2 + 2x + 37$.

Vervolgens moeten we ook $q(x)$ en $r(x)$ in de juiste vorm brengen, dus $q(x) = (x^2 + b)s(x) + t(x)$, met s en t monische polynomen van graad 1, en analoog $r(x)$. We vinden uiteindelijk dat $q(x) = (x^2 - 1)(x + 4) + x - 4$ en $r(x) = (x^2 + 1)(x - 11) + x + 48$. In totaal levert dat:

$$p(x) = (x^4 + 5)((x^2 - 1)(x + 4) + x - 4) + (x^2 + 1)(x - 11) + x + 48$$

b. Oplossen van de recurrente betrekking voor $A(k)$ (zie college 9):

$$A(k) = \begin{cases} 1 & k = 1 \\ 2A(k-1) + 2 & k > 1 \end{cases}$$

(1) Herhaald invullen

$$A(k) = 2A(k-1) + 2 = 2(2A(k-2) + 2) + 2 = 2^2A(k-2) + 2^2 + 2 = 2^2(2A(k-3) + 2) + 2^2 + 2 = 2^3A(k-3) + 2^3 + 2^2 + 2 = \dots = 2^\ell A(k-\ell) + 2^\ell + 2^{\ell-1} + \dots + 2^2 + 2.$$

Dit lijkt de algemene vorm te worden. We nemen nu $\ell = k - 1$, zodat we de beginvoorwaarde kunnen gebruiken. We krijgen dan:

$$= 2^{k-1}A(1) + (2^{k-1} + 2^{k-2} + \dots + 2^2 + 2 + 1) - 1 = 2^{k-1} + \sum_{i=0}^{k-1} 2^i - 1 = 2^{k-1} + 2^k - 1 - 1 = 3 \cdot 2^{k-1} - 2.$$

(2) Bewijs met inductie dat $A(k)$, gedefinieerd door de recurrente betrekking, inderdaad voor alle $k \geq 1$ gelijk is aan de in (1) gevonden functie:

- Basis: voor $k = 1$ klopt het: $3 \cdot 2^0 - 2 = 1 = A(1)$.

- Inductie-aanname: stel dat $A(k) = 3 \cdot 2^{k-1} - 2$ voor alle $k < K$.

- We moeten nu aantonen dat het dan ook klopt voor $k = K$, dus dat $A(K) = 3 \cdot 2^{K-1} - 2$. We gebruiken eerst de recurrente betrekking en daarna de inductie-aanname:
 $A(K) = 2A(K-1) + 2 = (\text{inductie-aanname}) 2 \cdot (3 \cdot 2^{K-2} - 2) + 2 = 3 \cdot 2^{K-1} - 2$. Klopt.
 Uitgedrukt in n is dit: $\frac{1}{2}(3n - 1)$.

c. Totaal aantal vermenigvuldigingen uitgedrukt in n : $\frac{n-1}{2} + \lg(n+1) - 1$. Dit is niet in tegenspraak met de afgeleide ondergrens van n vermenigvuldigingen omdat het hier gaat om polynomen die in een zeer speciale vorm gebracht zijn. Het in de juiste vorm brengen kost ook een zeker aantal $+$ / $-$ en $\times$ operaties. De stelling geldt voor willekeurige polynomen. Gebruikmakend van speciale eigenschappen van een polynoom kun je soms (hier dus ook) een sneller algoritme vinden.

Het totaal aantal, dus inclusief het preprocessen, is in dit geval de juist maat voor de complexiteit. Dat aantal moet met de n uit de stelling vergeleken worden. We zien dat je dan wel boven de ondergrens van n uitkomt (zie opgave 48). Alleen wanneer het polynoom heel vaak in verschillende punten moet worden geëvalueerd heeft het zin om preprocessing te doen. Dit hoeft namelijk maar één keer, waarna dan herhaald de snellere evaluatie kan worden toegepast.

opgave 48

a. Zie opgave 47 a. voor het eindantwoord.

Merk op dat het gegeven algoritme eigenlijk iets te veel doet. In regel 10 t/m 14 is het geval $t = j - 1$ namelijk niet nodig, aangezien q_{j-1} en r_{j-1} altijd 1 zijn omdat q en r monisch zijn. Vergelijk de berekening uit opgave 47 a. en sheet 12 van college 9.

c. We houden het algoritme uit de opgave aan en tellen het aantal optellingen. In regel (6) komt 1 optelling voor, door de while-loop uit regel (9) t/m (14) worden $j = 2^{k-1}$ optellingen gedaan voor $n = 2^k - 1$. Als $k = 1$ is $n = 1$, dus dan nul optellingen. Recurrente betrekking voor $A(k)$:

$$A(k) = \begin{cases} 0 & k = 1 \\ 2A(k-1) + 1 + 2^{k-1} & k > 1 \end{cases}$$

(1) Oplossing $A(k)$ van de recurrente betrekking vinden door herhaald invullen:

$$\begin{aligned} A(k) &= 2A(k-1) + 1 + 2^{k-1} = 2(2A(k-2) + 1 + 2^{k-2}) + 1 + 2^{k-1} = 2^2A(k-2) + 1 + 2 + 2 \cdot 2^{k-1} = \\ &= 2^2(2A(k-3) + 1 + 2^{k-3}) + 1 + 2 + 2 \cdot 2^{k-1} = 2^3A(k-3) + 1 + 2 + 2^2 + 3 \cdot 2^{k-1} = \dots = \\ &= 2^\ell A(k-\ell) + 1 + 2 + \dots + 2^{\ell-1} + \ell \cdot 2^{k-1}. \end{aligned}$$

In de hierboven afgeleide algemene vorm kiezen we nu $\ell = k - 1$, zodat we kunnen gebruiken dat $A(1) = 0$. We krijgen dan:

$$= 2^{k-1}A(1) + 1 + 2 + \dots + 2^{k-2} + (k-1) \cdot 2^{k-1} = 2^{k-1} - 1 + (k-1) \cdot 2^{k-1} = k \cdot 2^{k-1} - 1.$$

Nu met inductie bewijzen dat $k \cdot 2^{k-1} - 1$ de oplossing is:

- Basis: het klopt voor $k = 1$: $1 \cdot 2^0 - 1 = 0 = A(1)$.

- Inductie-aanname: stel dat $k \cdot 2^{k-1} - 1$ voor alle $k < K$ inderdaad de oplossing is van de recurrente betrekking, dus dat $A(k) = k \cdot 2^{k-1} - 1$ voor alle $k < K$ (dus voor alle $k \leq K - 1$).

- Nu aantonen dat het dan ook klopt voor $k = K$. We gebruiken weer eerst de recurrente betrekking (dus de definitie van $A(K)$) en daarna de inductie-aanname:

$$A(K) = 2A(K-1) + 1 + 2^{K-1} = 2((K-1) \cdot 2^{K-2} - 1) + 1 + 2^{K-1} = (K-1) \cdot 2^{K-1} - 2 + 1 + 2^{K-1} = K \cdot 2^{K-1} - 1. \quad \text{QED}$$

Uitgedrukt in n betekent dit dat het aantal optellingen gelijk is aan: $\frac{1}{2}(n+1) \lg(n+1) - 1$.
 Immers: $n = 2^k - 1 \implies 2^k = n + 1$ en $k = \lg(n + 1)$.

Analoog aan de recurrente betrekking voor $A(k)$ leiden we de recurrente betrekking af waar $V(k)$ (het aantal vermenigvuldigingen) aan voldoet:

$$V(k) = \begin{cases} 0 & k = 1 \\ 2V(k-1) + 2^{k-1} & k > 1 \end{cases}$$

Vinden/bewijzen van de oplossing hiervan is analoog aan dat voor $A(k)$. Hieronder is alleen het vinden van de oplossing verkort weergegeven.

$$V(k) = 2V(k-1) + 2^{k-1} = 2(2V(k-2) + 2^{k-2}) + 2^{k-1} = 2^2V(k-2) + 2 \cdot 2^{k-1} = \dots =$$

(algemene vorm en toewerken naar $V(1) = 0$) $2^{k-1}V(1) + (k-1)2^{k-1} = (k-1)2^{k-1}$.

Het aantal vermenigvuldigingen uitgedrukt in n wordt nu: $\frac{1}{2}(n+1)(\lg(n+1)-1)$. Merk op dat het aantal vermenigvuldigingen uit de preprocessing-stap en het aantal vermenigvuldigingen uit de evaluatie (opgave 47) samen $\frac{1}{2}(n+1)(\lg(n+1)-1) + \frac{n-1}{2} + \lg(n+1) - 1 \geq n+1$ (als $n \geq 3$) bedraagt.