

Opgaven werkcollege 2

Algoritmen en complexiteit

14 september 2015

1 Wakker worden

Opgave 1. a) Geef een gesloten formule voor de recurrente betrekking:

$$T(n) = \begin{cases} 1 & \text{als } n = 0, \\ 2T(n-1) + c & \text{als } n > 0. \end{cases}$$

b) Geef een θ -afschatting voor $T(n)$.

2 Vermenigvuldigen van polynomen

Een *polynoom* is een expressie van de vorm:

$$p = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0.$$

Hier noemen we x de *variabele* en $a_0, a_1, \dots, a_n$ de *coëfficiënten* van p . De *graad* van p is n . We nemen in het vervolg aan dat de coëfficiënten van onze polynomen gehele getallen zijn.

Opgave 2 (Klassiek algoritme). We analyseren de algoritme die je (waarschijnlijk) zelf gebruikt als je polynomen vermenigvuldigt. De algoritme is als volgt:

Algoritme 1.

Input: de coëfficiënten van $a = a_n x^n + \dots + a_1 x + a_0$ en $b = b_n x^n + \dots + b_1 x + b_0$.

Output: de coëfficiënten van $c = a \cdot b$.

1. for $i = 0, \dots, n$ do $d_i \leftarrow a_i x^i \cdot b$
2. return coëfficiënten $c = d_0 + d_1 + \dots + d_n$.

- a) Voer Algoritme 1 uit op $a = 5x^2 + 2x + 1$ en $b = x^2 + 3x + 5$.
- b) Hoeveel vermenigvuldigingen kost Algoritme 1 voor algemene a en b van graad n ? (Let op: 'vermenigvuldigingen' met machten van x tellen niet mee; de machten van x fungeren als indices in een array.)
- c) En hoeveel optellingen? (Ook hier telt 'optellen' van verschillende machten van x niet mee.)

- d) Geef een O -afschatting voor het totale aantal operaties (optellingen plus vermenigvuldigingen).

Opgave 3 (Verdeel en heers). We doen een poging om een snellere algoritme te maken door de verdeel-en-heers-strategie toe te passen. Zij a en b polynomen van graad kleiner dan $n = 2^k$ voor een $k \in \mathbb{N}$. Neem $m = n/2$. We splitsen a en b in twee delen

$$\begin{aligned}a &= a_1 x^m + a_0, \\ b &= b_1 x^m + b_0,\end{aligned}$$

met a_1, a_0, b_1, b_0 polynomen van graad kleiner dan m . Dan krijgen we

$$ab = a_1 b_1 x^n + (a_0 b_1 + a_1 b_0) x^m + a_0 b_0.$$

Bovenstaande geeft aanleiding tot een recursieve algoritme.

- Geef de recursieve algoritme.
- Welke recurrente betrekking hoort bij het aantal operaties dat de algoritme kost?
- Gebruik Akra-Bazzi om een θ -afschatting te vinden voor de algoritme. Hebben we iets bereikt?

Opgave 4 (Karatsuba's algoritme). Ons werk in Opgave 3 was niet voor niets, want een simpele observatie heeft een interessant gevolg. We houden dezelfde definities als in Opgave 3. De observatie is

$$ab = a_1 b_1 x^n + ((a_0 + a_1)(b_0 + b_1) - a_0 b_0 - a_1 b_1) x^m + a_0 b_0.$$

- Controleer de observatie.
- Wat is nu de recurrente betrekking voor het aantal operaties?
- Geef een θ -afschatting voor het aantal operaties.