Tentamen Complexiteit IBC028

Met uitwerkingen

28 juni 2016, 8.30 - 11.30 uur

Dit tentamen bestaat uit vijf opgaven die alle even zwaar tellen.

Het tentamen is een gesloten-boek-tentamen, dat wil zeggen dat er tijdens het tentamen geen gebruik mag worden gemaakt van het boek en/of aantekeningen.

Voor alle vragen geldt: motiveer uw antwoord.

Opgave 1.

a. De functie T is gegeven door $T(1) = 1$ en

$$T(n) = 2T(\lfloor n/2 \rfloor) + T(\lfloor 2n/3 \rfloor) + n^2$$

als $n > 1$. Bewijs dat $T(n) = O(n^2)$.

Uitwerking:

We bewijzen dat $T(n) \leq 18n^2$, met (sterke) inductie naar n. Voor $n = 1$ volgt dit uit $T(1) = 1$; voor $n > 1$ geldt dat $\lfloor n/2 \rfloor < n$ en $\lfloor 2n/3 \rfloor < n$ en mogen we dus op beide delen de inductiehypothese toepassen.

$$\begin{array}{rcll} T(n) & = & 2T(\lfloor n/2 \rfloor) + T(\lfloor 2n/3 \rfloor) + n^2 & \text{(gegeven)} \\ & \leq & 2 * 18\lfloor n/2 \rfloor^2 + 18\lfloor 2n/3 \rfloor^2 + n^2 & \text{(inductiehypothese)} \\ & \leq & 2 * 18(n/2)^2 + 18(2n/3)^2 + n^2 & \\ & = & 9n^2 + 8n^2 + n^2 & \\ & = & 18n^2, & \end{array}$$

waarmee het gevraagde is bewezen.

b. Een recursief algoritme met invoer van grootte n bestaat uit vier recursieve aanroepen met invoer van grootte $n/2$; de rest van het algoritme heeft tijdcomplexiteit $\Theta(n^2)$. Bepaal de tijdcomplexiteit van het algoritme.

Uitwerking:

Schrijf $T(n)$ voor de tijdcomplexiteit van het algoritme bij invoer van grootte n. Volgens het gegeven geldt dan

$$T(n) = 4T(n/2) + f(n)$$

voor $f(n) = \Theta(n^2)$. Volgens geval twee van de Master stelling geldt dan $T(n) = \Theta(n^2 \log n)$.

Opgave 2.

a. Geef een schets van een algoritme om van een ongesorteerde verzameling van n getallen de mediaan te bepalen, lineair in n.

Uitwerking:

We doen het iets algemener: we geven een algoritme om het k-de element van n gegeven getallen de bepalen voor als ze op volgorde zouden staan. De mediaan is dan deze waarde voor $k = n/2$.

Splits de n getallen op in $n/5$ groepjes van 5. Bepaal van elke van deze 5 de mediaan, in constante tijd. Bepaal recursief de mediaan M van al deze medianen. Tel hoeveel van alle n getallen kleiner zijn dan M. Afhankelijk van het antwoord kunnen uit de $n/10$ groepjes met mediaan $< M$, of juist de andere $n/10$ groepjes, elk drie elementen worden verwijderd. Laat het algoritme recursief los op de resterende $7n/10$ elementen. Schrijf $T(n)$ voor de tijdcomplexiteit van dit algoritme bij invoer van grootte n. Dan geldt

$$T(n) = T(n/5) + T(7n/10) + O(n),$$

waarmee met inductie kan worden bewezen dat $T(n) = O(n)$.

b. Is voor

$$T(n) = 2T(n/2) + n/\log n$$

de Master stelling van toepassing? Zo nee, geef aan waarom niet, zo ja, geef de resulterende functie f met $T(n) = \Theta(f(n))$.

Uitwerking:

voor geval (1) zou moeten gelden $n/\log n = O(n^{1-\epsilon})$ voor zekere $\epsilon > 0$. Voor geval (2) zou moeten gelden $n/\log n = \Theta(n)$. Voor geval (3) zou moeten gelden $n/\log n = \Omega(n^{1+\epsilon})$ voor zekere $\epsilon > 0$.

Geen van deze voorwaarden geldt, dus de Master stelling is niet van toepassing.

Opgave 3.

Gegeven zijn n punten in het platte vlak, en n rechthoeken met zijden parallel aan x- of y-as. Geef een $O(n \log n)$ algoritme dat met behulp van een sweepline vaststelt of tenminste een van de punten binnen tenminste een van de rechthoeken ligt.

Uitwerking:

Sorteer alle $3n$ x-waarden: de x-waarden van de punten, en van elke rechthoek de x waarde van de linker kant en de x waarde van de rechterkant. Deze worden met een sweepline doorlopen, van klein naar groot. Onderweg worden twee binaire zoekbomen bijgehouden: een voor de y-waarden van de onderkanten van alle snijdende rechthoeken en een voor de y-waarden van de bovenkanten van alle snijdende rechthoeken.

Als bij het doorlopen van de x- waarden een punt aan de orde is, wordt de bijbehorende y-waarde y_p in beide zoekbomen opgezocht, en als in beide zoekboom het aantal y-waarden groter dan y_p gelijk is, dan is er geen punt binnen een rechthoek gevonden, en anders wel (en kan het proces worden gestopt).

Als bij het doorlopen van de x- waarden een linkerkant van een rechthoek aan de orde is, worden de y-waarden van de bijbehorende boven- en onderkant aan de beide zoekbomen toegevoegd.

Als bij het doorlopen van de x- waarden een rechterkant van een rechthoek aan de orde is, worden de y-waarden van de bijbehorende boven- en onderkant uit de beide zoekbomen verwijderd.

Als toevoegen en verwijderen in de zoekbomen $O(\log n)$ is, zoals bij AVL bomen of rood-zwart bomen, dan is elke stap $O(\log n)$. Aangezien er $3n$ stappen zijn, is de totale complexiteit $O(n \log n)$.

Opgave 4.

a. Geef een voorbeeld van een 3-CNF die onvervulbaar (unsat) is.

Uitwerking:

Neem de 3-CNF bestaande uit de volgende 8 clauses:

$p \vee q \vee r$

$\neg p \vee q \vee r$

$p \vee \neg q \vee r$

$\neg p \vee \neg q \vee r$

$p \vee q \vee \neg r$

$\neg p \vee q \vee \neg r$

$p \vee \neg q \vee \neg r$

$\neg p \vee \neg q \vee \neg r.$

b. Bewijs dat $\leq_P$ transitief is.

Uitwerking:

Als $L_1 \leq_P L_2$ en $L_2 \leq_P L_3$ dan zijn er polynomiale f en g met

$$x \in L_1 \iff f(x) \in L_2$$

en

$$x \in L_2 \iff g(x) \in L_3.$$

Er geldt dus

$$x \in L_1 \iff f(x) \in L_2 \iff g(f(x)) \in L_3.$$

Dit bewijst $L_1 \leq_P L_3$ aangezien $g \circ f$ polynomiaal is omdat f en g beide polynomiaal zijn. Dit bewijst dat $\leq_P$ transitief is.

Opgave 5.

Voor een vaste n luidt het n-kleurbaarheidsprobleem (n-COLOR) als volgt:

Gegeven een graaf. Kunnen hiervan de knopen zodanig worden gekleurd met n kleuren zodanig dat voor elke kant de uiteinden een verschillende kleur hebben?

a. Geef aan wat er bewezen moet worden als we willen concluderen dat 4-COLOR NP-compleet is, gebruikmakend van het feit dat 3-COLOR NP-compleet is.

Uitwerking:

Er moet bewezen worden dat 4-COLOR in NP zit, en dat 3-COLOR $\leq_P$ 4-COLOR, oftewel, er moet een polynomiale f gevonden waarvoor geldt dat

$$x \in \text{3-COLOR} \iff f(x) \in \text{4-COLOR}.$$

b. Geef dit bewijs.

Uitwerking:

Voor het bewijs dat 4-COLOR in NP zit keizen we als certificaat de kleuring van een graaf met vier kleuren. Het algoritme om dit te checken doorloopt alle kanten en checkt of voor elke kant of de knopen aan de uiteinden een verschillende kleur hebben; dit kan in polynomiale tijd.

Voor het andere deel kiezen we voor f de transformatie die aan een graaf een knoop toevoegt en die knoop met alle bestaande knopen met een kant verbindt. Voor deze f bewijzen we

$$x \in \text{3-COLOR} \iff f(x) \in \text{4-COLOR}.$$

Als x een driekleuring heeft, dan geven we de nieuw knoop in $f(x)$ een vierde kleur, en geven we alle ander knopen in $f(x)$ de kleur die ze in x ook al hadden. Dit geeft een vierkleuring voor $f(x)$.

Omgekeerd, als we een vierkleuring van $f(x)$ hebben, komt de kleur van de nieuwe knoop verder niet voor omdat die met alle andere knopen verbonden is. Alle andere knopen zijn dus gekleurd met de resterende drie kleuren, waarmee we een driekleuring van x hebben.