Tentamen Complexiteit
Dinsdag 12 augustus 2008, 10.00-13.00 uur

Geef een *duidelijke* toelichting bij al je antwoorden!
Veel succes!

Opgave 1. (20 punten)
a. (5 punten)
Leg uit wat een beslissingsboom is voor algoritmen gebaseerd op arrayvergelijkingen: wat stellen de interne knopen en de bladeren voor, een pad van de wortel naar een blad, linker- en rechtersubboom van een knoop, de hoogte van de boom, etcetera.
b. (5 punten)
Bewijs met behulp van een beslissingsboomargument dat *elk* algoritme dat de *mediaan* (= middelste in grootte) van n verschillende elementen opspoort (m.b.v. arrayvergelijkingen) ten minste $\lceil \lg n \rceil$ vergelijkingen moet doen in de worst case.
c. (5 punten)
Laat zien dat het probleem van het vinden van de mediaan met maximaal $\Theta(n \lg n)$ arrayvergelijkingen *kan* worden opgelost.
d. (5 punten)
Volgt uit **b.** dat het algoritme uit **c.** niet-optimaal is? Licht je antwoord toe.

Opgave 2. (20 punten)
Gegeven een array A met n verschillende elementen $A[1], \ldots, A[n]$.
a. (2 punten)
Wat is een inversie?
b. (5 punten)
Bewijs: elk algoritme dat A sorteert met behulp van arrayvergelijkingen en dat na elke vergelijking ten hoogste één inversie opheft moet ten minste $\frac{1}{2}n(n-1)$ vergelijkingen doen in de worst case.
c. (3 punten)
Toon aan —zonder het aantal vergelijkingen daadwerkelijk te tellen— dat *Bubblesort* ten minste $\frac{1}{2}n(n-1)$ arrayvergelijkingen moet doen in de worst case.

Het volgende onderdeel staat volledig los van **a.** t/m **c.**
d. (10 punten)
Bewijs dat het aantal arrayvergelijkingen dat *Shellsort* met sprongafstanden (= stapgrootte) $n/3, n/9, n/27, \ldots, 9, 3, 1$, in de worst case doet $\Omega(n^2)$ is. Neem aan dat n een macht van 3 is.

Opgave 3. (20 punten)
Gegeven is een array A dat n verschillende getallen bevat, met $n \geq 0$ en n een drievoud.
a. (6 punten)
We kunnen de grootste, de op een na grootste en de op twee na grootste van deze n ($n \geq 3$ een drievoud) waarden bepalen door van links naar rechts door het array te lopen en steeds groepjes van 3 te bekijken. Dit leidt tot een algoritme dat (in de worst case) $2n-3$ vergelijkingen doet. Beschrijf het algoritme (in woorden is voldoende) en laat zien dat het inderdaad $2n-3$ vergelijkingen doet in het slechtste geval.

b. (4 punten)
Bekijk het volgende *recursieve* algoritme dat het array oplopend sorteert. Bepaal eerst de drie grootste waarden en zet die meteen op hun goede plaats. Sorteer dan de rest van het array op dezelfde manier (*recursief*). Noem het aantal arrayvergelijkingen dat dit algoritme doet in de worst case $W(n)$. Leg uit waarom $W(n)$ voldoet aan de volgende recurrente betrekking:

$$W(n) = \begin{cases} 0 & n = 0 \\ W(n-3) + 2n - 3 & n \geq 3,\ n = 3k \text{ (n een drievoud)} \end{cases}$$

c. (10 punten)
Los de recurrente betrekking uit **b.** op door deze herhaald in zichzelf in te vullen en bewijs met behulp van volledige inductie dat de aldus gevonden oplossing (uitgedrukt in n) inderdaad voldoet. Laat bij het herhaald invullen de termen met n, $n-3$, $n-6$, etc. gewoon staan en veeg de 3-en bij elkaar. Dan is de algemene vorm beter te herkennen. Gebruik ten slotte dat $\sum_{i=1}^{n/3} i = \frac{n}{6}(\frac{n}{3} + 1)$.

Opgave 4. (25 punten)
Gegeven is een array A met $n = 2^k$ ($k \geq 1$, geheel) verschillende gehele getallen $A[1]$, $A[2]$, $\ldots$, $A[n]$. We bekijken een algoritme dat het grootste *en* het op een na grootste getal uit het array oplevert.

```
(1)   i := 1;
(2)   while i < n do
(3)       if A[i] > A[i + 1] then
(4)           wissel(A[i], A[i + 1]);
(5)       fi
(6)       i := i + 2;
(7)   od
(8)   d := 2;
(9)   while d < n do
(10)      i := d; j := i + d − 1;
(11)      while i ≤ n do
(12)          if A[i] > A[i + d] then
(13)              wissel(A[i], A[i + d]);
(14)              j := i − 1;
(15)          fi
(16)          if A[i] > A[j] then
(17)              wissel(A[i], A[i + d − 1]);
(18)          else if j ≠ i + d − 1 then
(19)                  wissel(A[j], A[i + d − 1]);
(20)              fi
(21)          fi
(22)          i := i + 2 ∗ d; j := i + d − 1;
(23)      od
(24)      d := 2 ∗ d;
(25)  od
```

De functie *wissel* verwisselt de waarden van zijn argumenten. Na afloop zitten in $A[n-1]$

en $A[n]$ het op een na grootste respectievelijk het grootste van de oorspronkelijke array-elementen. Het algoritme bekijkt steeds twee opeenvolgende stukken van het array ter lengte d, waarbij d telkens een 2-macht is.

a. (4 punten)
Neem aan dat het algoritme (voor $n = 16$) regel (1) t/m (8) reeds heeft gedaan, en dat de inhoud van het array A nu als volgt is:

$$9, 11, 7, 13, 3, 4, 18, 21, 6, 20, 10, 12, 15, 16, 5, 14.$$

Het algoritme gaat met $d = 2$ verder bij regel (9). Voer het algoritme verder uit en geef als tussenstappen de inhoud van het array A steeds wanneer regel (22) net is uitgevoerd.
b. (7 punten)
Laat zien dat het aantal arrayvergelijkingen (regel (3), (12) en (16) samen) maatgevend is voor de complexiteit van het algoritme. Doe dit door het aantal keren dat een regel wordt gedaan in verband te brengen met het aantal keren dat de test in regel (3) of (12) of (16) wordt uitgevoerd.

We bekijken vanaf nu het totaal aantal verwisselingen dat het algoritme doet.
c. (8 punten)
Toon voor algemene n aan dat een oplopend gesorteerd rijtje een best case geval is, en een aflopend gesorteerd rijtje een worst case geval. Geef een duidelijke uitleg.
d. (6 punten)
Geef voor $n = 8$ nog een ander, concreet voorbeeld van een best case invoerrijtje. Geef voor $n = 8$ eveneens een ander, concreet voorbeeld van een worst case invoerrijtje. Motiveer duidelijk je antwoord.

Opgave 5. (15 punten)
a. (5 punten)
We bekijken het probleem DNF-SAT: Gegeven een logische formule ϕ in DNF (*disjunctieve* normaalvorm). Bestaat er een waardering van de in ϕ voorkomende logische variabelen die ϕ waar maakt? Laat zien dat dit probleem in $\mathcal{P}$ zit.
Merk op: een formule staat in disjunctieve normaalvorm als hij bestaat uit een disjunctie ($\vee$) van conjuncties ($\wedge$) van literals.
b. (5 punten)
Leg uit wat een polynomiale reductie is en bewijs: als $P \leq_P Q$ en Q zit in $\mathcal{P}$, dan zit P ook in $\mathcal{P}$.
c. (5 punten)
Hieronder is SAT het uit het college bekende NP-volledige probleem: Gegeven een logische formule ϕ in *conjunctieve* normaalvorm. Bestaat er een waardering van de in ϕ voorkomende logische variabelen die ϕ waar maakt?

(i) Bestaat er een polynomiale reductie van SAT naar DNF-SAT? Motiveer je antwoord.
(ii) Bestaat er een polynomiale reductie van DNF-SAT naar SAT? Motiveer je antwoord.

Er wordt hier alleen naar het bestaan gevraagd. Je hoeft dus geen reductie te geven.

Veel succes!