**Tentamen Algoritmen en Datastructuren**
**3 februari 2004**

Tijdsduur 3 uur. Voorzie alle in te leveren bladen van je naam, en nummer ze. Schrijf op het eerste blad het aantal ingeleverde bladen. Werk netjes, formuleer scherp en zorgvuldig. Schrijf duidelijk leesbaar.

Je mag ervan uitgaan, dat een efficiente implementatie van een priority queue beschikbaar is. Er is ook een methode *FFT* beschikbaar, die handig gebruikt kan worden om middels convolutie het product van twee polynomen (veeltermen) efficient te berekenen. Je hoeft deze bouwstenen dus niet te implementeren, maar kunt ze gebruiken. Je moet bij gebruik wel de verwachte complexiteit aangeven.

**Opgave 1** (20%). (a) Geef de definitie van een AVL-boom.
(b) Bewijs dat een AVL-boom met $n$ knopen een hoogte heeft van de orde $\mathcal{O}(\log n)$.

**Opgave 2** (25%). (a) Geef de definitie van een minimum spanning tree in een samenhangende gewogen ongerichte graaf.
(b) Geef een algoritme in pseudocode om zo'n minimum spanning tree te bepalen.

**Opgave 3** (30%). Gegeven is een flow netwerk $N = (V, c, s, t)$ met verzameling $V$ van knopen, capaciteitsfunctie $c$, source $s$, en target $t$. Het aantal knopen is $n$, het aantal verbindingen is $m$. De capaciteit van een pad in de graaf is het minimum van de capaciteiten van de verbindingen.
(a) Geef een $\mathcal{O}((n+m)\log n)$ algoritme in pseudocode ter bepaling van een pad van $s$ naar $t$ met maximale capaciteit.
(b) Laat zien dat je algoritme de gewenste efficientie heeft.

**Opgave 4** (25%). Gegeven zijn een reëel getal $a$ en een rij van $n$ reële getallen $z_0, \ldots, z_{n-1}$, verschillend van $a$ en van elkaar. Gevraagd: de rij van coëfficienten van de unieke polynoom $p(x)$ van de graad $n$ die de waarde 1 in het punt $a$ heeft en 0 is in alle punten $z_i$.
Aanwijzing. Voor $n = 0$ voldoet de polynoom $p(x) = 1$. Voor $n = 1$ voldoet $p(x) = (a - z_0)^{-1}(x - z_0)$. Je hoeft de invoer niet te controleren.
(a) Geef hiertoe een verdeel- en heersalgoritme. Het dient een tijdscomplexiteit $\mathcal{O}(n \log^2 n)$ te hebben.
(b) Laat zien dat je algoritme de gewenste efficientie heeft.