Eerste deeltoets Algoritmiek

5 maart 2012, 8.30 – 10.30, Educ-α.

Alles past op een (dubbel) blad; beantwoord de vragen *kort*!

1. **Subsitututie-methode:** Bewijs met de Substitutiemethode dat de oplossing van $T(n) = T(n/2) + d.n$ voldoet aan $T(n) = O(n)$.

2. **Master Theorem:** Wat is de oplossing van de recurrentie $T(n) = 2.T(n/4) + n$?

3. **De Top-T:** Gegeven: een ongesorteerde rij A van n getallen, en een integer $T \leq n$. Gevraagd: de T grootste getallen uit A, in oplopende volgorde. Geef een zo snel mogelijk algoritme dat de gevraagde getallen bepaalt. Wat is de tijdcomplexiteit van je algoritme?

4. **Stokzagen met knoesten:** Een stokjesfabriek verzaagt een lat van (integer) lengte n in een of meer stokjes, waarbij een stokje van lengte i wordt verkocht voor p_i. De lat kan op elke integer-positie s worden gezaagd, maar er zitten onregelmatige knoesten in, waardoor zagen op positie s, zaagkosten z_s kost. De *revenu* van de lat is de opbrengst van de stokjes min de totale zaagkosten.

   (a) Geef een recursieve uitdrukking voor de revenu van een lat, waarbij je als topkeuze hanteert: wat is de positie van de hoogste zaagsnede.

   (b) Wat is de looptijd van het resulterende DP-algoritme?

5. **Beladingen tellen:** Een smokkelaar heeft een rugzak van omvang M, keuze uit n objecten 0 t/m $n-1$ van gewicht w_0 t/m w_{n-1} en winst p_0 t/m p_{n-1}. De smokkelaar wil weten op hoeveel manieren hij zijn rugzak kan beladen. Laat zien hoe je het *aantal* deelverzamelingen met gewicht $\leq M$ kunt bepalen in $O(n.M)$ tijd.

6. **De Lift:** Een schooljuf komt met n kinderen, met gewicht a_0 t/m a_{n-1}, bij een lift met laadvermogen L. Juf wil *zo veel mogelijk* kinderen in de lift zetten, en met de andere kinderen de trap nemen. Ze komt op het idee, de kinderen van *licht naar zwaar* in de lift te laten lopen tot het volgende kind er niet meer bij kan.

   Bewijs dat de juf op deze manier een zo groot mogelijk aantal kinderen in de lift krijgt.

7. **Fibonacci-Huffman:** Geef een optimale Huffman-code voor $\{a, b, c, d, e, f, g, h\}$, waarbij de relatieve frequenties zich verhouden als de Fibonacci-getallen:

σ	a	b	c	d	e	f	g	h
$f(\sigma)$	1	1	2	3	5	8	13	21

8. **Union-Find Implementatie:** De *woud-representatie* is een manier om verzamelingen (sets) bij te houden die je kunt samenvoegen. Welke twee verbeteringen moet je toevoegen aan de woud-representatie om te komen tot de snelst bekende implementatie van dit probleem? Noem ze en geef van elk in één zin aan hoe ze werken.

9. **Gemiddelden:** De *gemiddelde* (average), *expected* (verwachte) en *geamortiseerde* amortized complexity berekenen alledrie een gemiddelde over looptijden.

   Waarover wordt gemiddeld bij het berekenen van een verwachte, een gemiddelde, en een geamortiseerde complexiteit?

10. **Amortisering:** Op een datastructuur wordt een reeks stappen 1 t/m n uitgevoerd. Stap i kost 1 als i geen tweemacht is, en $i+1$ is als i wel een tweemacht is.

    Bewijs dat de geamortiseerde kosten hoogstens 3 per stap zijn.