

Deeltoets 1 Algoritmiek

23 december 2008, 14.00 – 16.00 uur, BBL 420.

- Zoeken in een graaf:** Gegeven zijn een gerichte graaf $G = (V, E)$ en twee knopen s en t in de graaf. Geef een algoritme (pseudocode) dat in $O(n + m)$ tijd bepaalt of er een pad van s naar t bestaat, en zo ja, zo'n pad oplevert.
 Waarom is je algoritme correct en waarom voldoet het aan de tijdgrens?
- Greedy algoritme: Films op USB stick.** Een film liefhebber heeft n filmbestanden, met bestandsgrootten s_1 tot en met s_n , en een USB stick met capaciteit M (alles in MB). Hij wil *zoveel* mogelijk films op zijn stick zetten.
 (a) Omschrijf dit als een optimaliseringsprobleem (zoekruimte, etc.).
 De liefhebber sorteert zijn films oplopend naar grootte, en zet de films in die volgorde op de USB stick, tot een film niet meer past.
 (b) Wat betekenen GCP en OSP en formuleer deze twee eigenschappen voor de omschreven strategie.
 (c) Bewijs de GCP.
- Amortisering:** Er wordt een reeks van n operaties uitgevoerd, waarbij de prijs van de i^e operatie 1 is als i géén tweemacht, en $i + 1$ is als i wel een tweemacht is. Bereken de geamortiseerde kosten per operatie. Gebruikt je berekening de Verzamel (aggregate), Krediet (accounting), of Potentiaal (potential) methode?
- Union-Find structuur.** Een union-find structuur representeert elke verzameling als een boom en gebruikt gewogen union: dus bij een union wordt steeds de wortel van de lichtste boom een kind van de wortel van de andere boom gemaakt. Toon aan dat de lengte van het pad van een willekeurig element naar de wortel van zijn verzameling nooit meer dan $\log n$ kan zijn (n is het aantal elementen).
- Minimum Uitputting van Avatar.**

Je programmeert een avatar die een vierkant speelveld van n bij n cellen moet oversteken. Een stap van de avatar gaat vooruit (dwz., naar de volgende rij), maar kan recht of schuin (dwz., in dezelfde of naastliggende kolom). De begin- en eindpositie zijn vrij te kiezen in de onderste, cq bovenste rij. Het bezoeken van positie (i, j) kost $C[i, j]$. Geef een algoritme dat in $O(n^2)$ tijd een lijst geeft van de posities op een goedkoopst pad.

	1	2	3	4
4	2	2	4	6
3	3	5	5	2
2	2	4	1	0
1	2	3	2	10

In het voorbeeld is de goedkoopste route (kosten 8) vet gedrukt. De uitvoer van het algoritme zou hier bv zijn: $(1, 3)$ $(2, 4)$ $(3, 4)$ $(4, 3)$