

Algoritmiëk (INFOAL) 18 december 2006

N.B. U mag tijdens het tentamen maximaal 4 kantjes A4 eigen aantekeningen raadplegen. Deze dient U met Uw werk in te leveren, en kan U later weer ophalen.

Elk van de vragen 1, 2a, 2b, 3 en 4 telt voor twee punten.

Veel succes!

Opgave 1: Grafen

Een ongerichte graaf $G = (V, E)$ heet *kubisch* als elke knoop in G graad precies drie heeft. Hoeveel tijd kost het om te testen of een graaf kubisch is, als de graaf gegeven is met de adjacency-list datastructuur? Hoeveel tijd kost het om te testen of een graafkubisch is, als de graaf gegeven is met de adjacency matrix datastructuur? Licht Uw antwoorden kort maar helder toe.

Opgave 2: Dynamisch Programmeren

Gegeven zij een verzameling van n positieve gehele getallen $A = \{a[1], \dots, a[n]\}$ en een positief geheel getal B . We willen bepalen hoeveel deelverzamelingen van A som precies gelijk aan B hebben. Dit doen we met dynamisch programmeren.

We schrijven: $M(i, C)$ als het aantal deelverzamelingen van $\{a[1], \dots, a[i]\}$ met som precies C .

- Geef een recurrente betrekking voor $M(i, C)$. Vergeet de basisgevallen niet.
- Geef een dynamisch programmeer algoritme dat het probleem oplost. Uw algoritme moet in $O(nB)$ tijd werken.

In het geval het U niet lukte om een recurrente betrekking op te stellen in onderdeel a), dan mag U de volgende (op zich incorrecte) betrekking gebruiken.

$$\begin{aligned} M(0, C) &= 0 \text{ voor alle } C \in \mathbb{N} \\ M(i, C) &= \begin{cases} M(i-1, C) + 1 & \text{als } a_i > C \\ \max\{M(i-1, C), M(i, C - a_i) + 1\} & \text{als } a_i < C \end{cases} \end{aligned}$$

Opgave 3: Greedy algorithms

Stel P_1, \dots, P_n zijn n programma's die moeten worden opgeslagen op een disk. Programma P_i kost s_i kilobytes opslagruimte, en de capaciteit van de disk is D kilobytes, met $D < \sum_{i=1}^n s_i$.

Stel we willen het aantal programma's op de disk maximaliseren. Bekijk het volgende greedy algoritme: kies programma's in volgorde van niet-dalende waarde s_i , totdat ze niet meer erbij passen. Geeft dit algoritme een optimale oplossing?

Bewijs Uw antwoord.

Opgave 4: Divide and conquer / Simplification

Stel we hebben twee gesorteerde rijen integers $A[1 \dots n]$ en $B[1 \dots n]$. Elke rij integers heeft precies n verschillende getallen, en is gesorteerd in stijgende volgorde.

We willen een algoritme dat het volgende probleem oplost: we zoeken indices $i, j \in \{1, \dots, n\}$ zodat $A[i] + B[j] = i$.

Geef een algoritme dat dit probleem in $O(n)$ tijd oplost. Waarom is Uw algoritme correct?