Toets Algoritmiek 2003

U heeft 1 uur en 45 minuten de tijd voor deze toets.

1. (2 punten) Een ongerichte graaf $G = (V, E)$ heet *3-regulier* als elke knoop in G graad precies drie heeft. Hoeveel tijd kost het om te testen of een graaf 3-regulier is, als de graaf gegeven is met de adjacency-list datastructuur? Hoeveel tijd kost het om te testen of een graaf 3-regulier is, als de graaf gegeven is met de adjacency matrix datastructuur? Licht je antwoorden kort maar helder toe.

2. (4 punten) Gegeven zij een verzameling van n positieve gehele getallen $A = \{a_1, \ldots, a_n\}$ 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 $\{a_1, \ldots, a_i\}$ met som precies C.

   (a) Leg uit dat $M(0, C) = 0$ als $C > 0$. Leg uit dat $M(0, 0) = 1$. Leg uit dat $M(i, C) = M(i-1, C)$ als $a_i > C$.

   (b) Stel $i \geq 1$. Precies één van de volgende recurrente betrekkingen is correct. Welke van de betrekkingen is de juiste? Leg *duidelijk* uit waarom die betrekking juist is.

$$M(i,C) = \begin{cases} M(i-1,C) & \text{als } a_i > C \\ \max\{M(i-1,C), M(i,C-a_i)\} & \text{als } a_i \leq C \end{cases} \quad (1)$$

$$M(i,C) = \begin{cases} M(i-1,C) & \text{als } a_i > C \\ M(i-1,C) + M(i,C-a_i) & \text{als } a_i \leq C \end{cases} \quad (2)$$

$$M(i,C) = \begin{cases} M(i-1,C) & \text{als } a_i > C \\ M(i-1,C) + M(i-1,C-a_i) & \text{als } a_i \leq C \end{cases} \quad (3)$$

$$M(i,C) = \begin{cases} M(i-1,C) & \text{als } a_i > C \\ \max\{M(i-1,C), M(i-1,C-a_i)\} & \text{als } a_i \leq C \end{cases} \quad (4)$$

   (c) Geef een dynamisch programmeer algoritme dat het probleem oplost. Uw algoritme moet in $O(nB)$ tijd werken. (U mag bij dit onderdeel er vanuitgaan dat de recurrente betrekking die U bij het vorige onderdeel gekozen heeft de juist is.)

   (d) Geef een dynamisch programmeer algoritme dat het probleem oplost, en dat $O(nB)$ tijd en $O(B)$ geheugen gebruikt.

3. (2 punten) Stel we willen het single pair shortest paths probleem oplossen, en we zoeken naar de afstand van s naar v.

   (a) Klopt het dat we kunnen stoppen met Dijkstra's algoritme zodra v uit C gehaald is? Waarom?

   (b) Stel we stoppen met Dijkstra's algoritme zodra v uit C gehaald is, en we implementeren hierbij de priority queue met behulp van een heap. Wat is nu de worst case tijd van het algoritme?

4. (2 punten) Stel we hebben een array (niet noodzakelijk gesorteerd) A met n gehele getallen $A[1] \cdots A[n]$. Ook is gegeven een geheel getal $x \in \mathbf{Z}$. Geef een algoritme dat in $O(n \log n)$ tijd test of er i, j zijn, $1 \leq i \leq j$ met $A[i] + A[j] = x$.