Toets Algoritmiek 2002

U heeft 1 uur en 45 minuten de tijd voor deze toets. Schrijf Uw naam op alle blaadjes die U inlevert. Elk van de opgaves 1, 2, 3 en 4 telt even zwaar. Veel succes!

1. Leg in eigen woorden uit hoe Strassen's algoritme voor matrixvermenigvuldiging werkt. Hierbij mag U zonder verder bewijs gebruiken dat twee 2 bij 2 matrices met zeven vermenigvuldigingen gedaan kunnen worden, en uitgaan van het geval dat de matrices n bij n matrices zijn met n een tweemacht.

2. We beschouwen het volgende probleem. Gegeven zijn M verzamelingen positieve gehele getallen, $A_1, \ldots, A_M$, en een getal B. Kunnen we uit elke verzameling A_i, $1 \leq i \leq M$ precies één getal nemen, zodat de som van deze (M) getallen precies B is. (Bijvoorbeeld: als $M = 3$, $A_1 = \{1, 2, 3\}$, $A_2 = \{10, 20\}$, and $A_3 = \{100, 200\}$, $B = 111$, dan is het antwoord ja: neem 1, 10, 111. Als $B = 1$, dan is het antwoord nee. Met andere woorden, we zoeken getallen $a_1 \in A_1$, …, $a_M \in A_M$ met $\sum_{i=1}^{M} = B$.)

   Laat voor $0 \leq i \leq M$, $0 \leq j \leq B$, $P(i,j)$ true zijn als uit elke verzameling $A_1, \ldots, A_i$ precies een getal genomen kan worden zodat de som van deze (i) getallen precies j is.

   We hebben de volgende uitdrukkingen voor $P(i,j)$: $P(0,0) = $ true. Voor $j > 0$, $P(0,j) = $ false. Als $i > 0$ dan

   $$P(i,j) = \bigvee_{x \in A_i, x \leq j} P(i-1, j-x)$$

   (Deze betrekkingen hoeft U niet te bewijzen maar mag U als gegeven beschouwen.)

   Schrijf $C = \max_{1 \leq i \leq M} |A_i|$.

   Geef een algoritme dat in $O(MBC)$ tijd het probleem oplost, d.w.z., bepaalt of er getallen $a_1 \in A_1$, …, $a_M \in A_M$ bestaan met $\sum_{i=1}^{M} a_i = B$.

3. Laat V een verzameling van n voorwerpen zijn, allemaal met een bepaalde waarde. De waarde van een voorwerp v wordt aangegeven met $w(v)$. Elke waarde is een positief geheel getal. We willen V in drie deelverzamelingen, V_1, V_2, en V_3, opdelen op een eerlijke manier. Preciezer

1

gezegd, we kijken of er deelverzamelingen V_1, V_2, en V_3 bestaan die een opsplitsing vormen van V (d.w.z. $V = V_1 \cup V_2 \cup V_3$ en $V_1 \cap V_2 = V_1 \cap V_3 = V_2 \cap V_3 = \emptyset$) en waarvoor geldt:

$$\sum_{v \in V_1} w(v) = \sum_{v \in V_2} w(v) = \sum_{v \in V_3} w(v).$$

Merk op dat er een eerlijke opsplitsing is dan en slechts dan als er twee (disjuncte) deelverzamelingen V_1 en V_2 zijn die elk een totale waarde $W/3$ hebben. Definieer nu voor $1 \leq i \leq n$, $0 \leq W_1 \leq W/3$ en $0 \leq W_2 \leq W/3$

$$M[i, W_1, W_2] = \begin{cases} \textbf{true} & \text{als } \{v_1, \ldots, v_i\} \text{ twee disjuncte deelverzamelingen heeft met } W_1 \text{ resp. } W_2 \text{ als totale waarde} \\ \textbf{false} & \text{anders} \end{cases}$$

Er is dus een eerlijke opsplitsing van V als $M[n, W/3, W/3] = \textbf{true}$. Geef een recursieve definitie van $M[i, W_1, W_2]$.

4. Gegeven n skiërs met hoogte $h_1, \ldots, h_n$, en n paar skies met hoogte $s_1, \ldots, s_n$. Het probeem is om aan iedere skiër een paar skies te geven zodat het gemiddelde verschil in hoogte van de skiër en de ski geminimaliseerd wordt. Dat wil zeggen als de i^{de} skiër de $p(i)^{de}$ ski krijgt willen we minimaliseren

$$\frac{1}{n} \sum_{i=1}^{n} |h_i - s_{p(i)}|$$

(i) Beschouw het volgende greedy algoritme. Zoek de skiër en het paar skies met minimaal hoogte verschil. Geef de skiër dit paar skies met minimaal hoogte verschil. Herhaal dit proces totdat alle skiërs een paar skies hebben. Bewijs dat het algoritme correct is of geef een tegenvoorbeeld.

(ii) Beschouw het volgende greedy algoritme. Geef de kortste skiër het kortste paar skies, de op een na kortste skiër het op een na kortste paar skies, de op twee na kortste skiër de op twee na kortste paar skies, enzovoort. Bewijs dat het algoritme correct is of geef een tegenvoorbeeld.

Hint: Van bovenstaande algoritmes is één algoritme niet correct en één algoritme correct.

2