Dit tentamen is in elektronische vorm beschikbaar gemaakt door de $\mathcal{BC}$ van A–Eskwadraat. A–Eskwadraat kan niet aansprakelijk worden gesteld voor de gevolgen van eventuele fouten in dit tentamen.

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. In een array zijn de salarisgegevens opgeslagen van de werknemers van een bedrijf. De afdeling personeelszaken wil graag weten voor welk bedrag x geldt dat precies 20 procent van alle werknemers tenminste x euro verdient. Kent U een algoritme om dit probleem op te lossen, waarvan de verwachte tijd lineair is in het aantal werknemers? Zo ja, leg uit hoe zo'n algoritme werkt.

2. Gegeven zijn twee strings, $X = x_1x_2\cdots x_n$ en $Y = y_1y_2\cdots y_m$. We willen de lengte van de langste gemeenschappelijk subsequence van X en Y vinden. Hiertoe definieren we voor $0 \leq i \leq n$ en $0 \leq j \leq m$:

   $M[i,j]$ = de lengte van de langste gemeenschappelijke subsequence van $x_1x_2\cdots x_i$ en $y_1y_2\cdots y_j$.

   (Voor $i = 0$ is $x_1x_2\cdots x_i$ per definitie de lege string, en voor $j = 0$ is $y_1y_2\cdots y_j$ per definitie de lege string.) We kunnen de volgende recurrente betrekking opstellen voor $M[i,j]$:

$$M[i,j] = \begin{cases} M[i-1,j-1]+1 & \text{als } i>0,\ j>0, \text{ en } x_i = y_j \\ \max(M[i-1,j], M[i,j-1]) & \text{als } i>0,\ j>0, \text{ en } x_i \neq y_j \\ 0 & \text{als } i=0 \text{ of } j=0 \end{cases}$$

   Geef een algoritme dat in $O(nm)$ tijd de lengte van de langste gemeenschappelijke subsequence van X en Y berekent. (N.B. Je hoeft de recurrente betrekking hierboven dus niet aan te tonen; je mag wel aannemen dat deze geldig is.)

3. 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 $\sum_{i=1}^{M} a_i = B$.)

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.

(i) Wat is $P(0,0)$ en $P(0,j)$ met $j > 0$?

(ii) Geef een recurrente betrekking voor $P(i,j)$, $i > 0$, $j \geq 0$. (Hint: kijk naar de verschillende getallen die in A_i zitten en druk $P(i,j)$ uit met behulp van een aantal termen van de vorm $P(i-1, \cdots)$.)

4. Gegeven een verzameling $S = \{(x_i, y_i) | 1 \leq i \leq n\}$ van tijdsintervallen. Denk bij zo'n interval aan een verzoek om een collegezaal voor een bijeenkomst van tijd x_i tot tijd y_i te reserveren. Minimaliseer het benodigde aantal collegezalen voor alle bijeenkomsten.

   Hierbij moet aan ieder verzoek voldaan worden en mogen er niet twee bijeenkomsten op hetzelfde tijdstip in dezelfde collegezaal gegeven plaats vinden.

   (i) Beschouw het volgende greedy algoritme. Plan zoveel mogelijk bijeenkomsten in de eerste ruimte (dit kan met een greedy algoritme, maar dat hoef je niet te laten zien). Plan daarna zoveel mogelijk bijeenkomsten in de tweede ruimte, dan zoveel mogelijk bijeenkomsten in de derde ruimte, enzovoort. Bewijs dat het algoritme correct is of geef een tegenvoorbeeld.

   (ii) Beschouw het volgende greedy algoritme. Verwerk de bijeenkomsten naar toenemende begintijd. Neem aan dat je bijeenkomst B plant. Als er een collegezaal C is, die al gebruikt is voor een andere bijeenkomst en als C aan B kan worden toegewezen zonder overlap met al eerder geplande bijeenkomsten in C, reserveer dan collegezaal C voor bijeenkomst B. Anders plan de bijeenkomst in een nieuwe collegezaal.

   Bewijs dat het algoritme correct is of geef een tegenvoorbeeld.

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

2