

Antwoorden opgaven werkcollege 8

Opgave 43

Zie uitwerking website

Opgave 44

We bekijken het aantal arrayvergelijkingen dat Shellsort doet op een gesorteerd rijtje. Daarbij gebruiken we Shell's sprongafstanden, met Insertion sort om de deelrijtjes te sorteren. Merk op dat Insertion sort op een gesorteerd rijtje ter lengte m precies $m - 1$ vergelijkingen doet en het rijtje onveranderd laat.

Allereerst wordt het rijtje $\frac{n}{2}$ -gesorteerd. Dat houdt in dat we Insertion sort doen op $\frac{n}{2}$ rijtjes van lengte 2. Aangezien de invoer gesorteerd is, is ieder deelrijtje dat ook. Hier doen we dus $\frac{n}{2} * (2 - 1) = \frac{n}{2}$ vergelijkingen. Vervolgens wordt het rijtje $\frac{n}{4}$ -gesorteerd, waarbij we $\frac{n}{4}$ (gesorteerde) rijtjes van lengte 4 sorteren, dus in totaal $\frac{n}{4} * (4 - 1) = \frac{3n}{4}$ vergelijkingen. Over het algemeen: in ronde i sorteren we $\frac{n}{2^i}$ (gesorteerde) rijtjes van lengte 2^i , voor een totaal aantal vergelijkingen van:

$$\frac{n}{2^i}(2^i - 1) = n - \frac{n}{2^i} = n - 2^{k-i}$$

We doen $k = \lg n$ sorteerrondes: $n/2$ -sorteren, $n/4$ -sorteren, $\dots$, 1-sorteren. Samen wordt dat:

$$\begin{aligned} \sum_{i=1}^k (n - 2^{k-i}) &= n \lg n - \sum_{i=1}^k 2^{k-i} = n \lg n - (1 + 2 + 2^2 + \dots + 2^{k-1}) \\ &= n \lg n - (2^k - 1) = n \lg n - n + 1 \in O(n \lg n) \end{aligned}$$

vergelijkingen

Opgave 45

a. We doen Counting sort op het rijtje 7, 1, 3, 1, 2, 4, 5, 7, 2, 4, 3. Met subscripts die gelijke waarden nummeren: $7_a, 1_a, 3_a, 1_b, 2_a, 4_a, 5_a, 7_b, 2_b, 4_b, 3_b$. We hebben $k = 7$ en initialiseren C op 0. Nadat we de eerste keer door het rijtje hebben gelopen ziet C er als volgt uit: 2, 2, 2, 2, 1, 0, 2. Vervolgens transformeren we C door een lopende som te nemen ($C[i] = C[i - 1] + C[i]$ voor elke i): 2, 4, 6, 8, 9, 9, 11.

$C[i]$ geeft nu aan hoeveel getallen er in A kleiner dan of gelijk aan i zijn; met andere woorden: $C[A[j]]$ geeft de meest rechter mogelijke positie van array-element $A[j]$ aan.

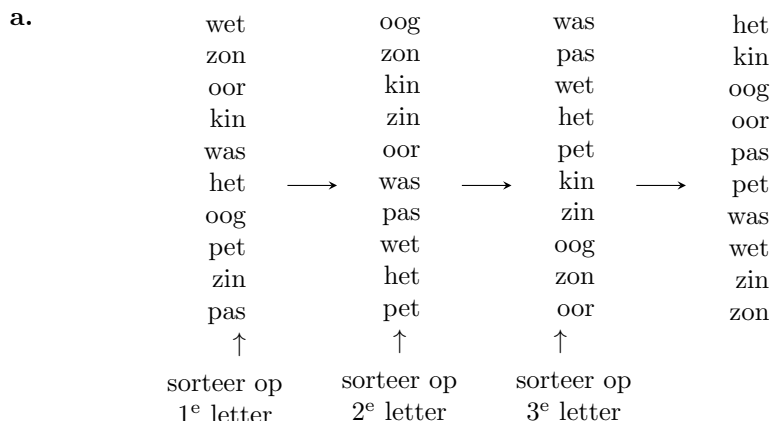
We plaatsen ten slotte voor iedere waarde $A[j]$ van het array (*van rechts naar links*) het getal op de *meest rechtse* positie (gegeven door $C[A[j]]$) en schuiven die positie één plaats naar links door $C[A[j]]$ met 1 te verlagen.

In volgorde krijg je dan de volgende toekenningen: $B[6] = 3_b, B[8] = 4_b, B[4] = 2_b, B[11] = 7_b, B[9] = 5, B[7] = 4_a, B[3] = 2_a, B[2] = 1_b, B[5] = 3_a, B[1] = 1_a, B[10] = 7_a$.

Uiteindelijk hebben we in B het rijtje $1_a, 1_b, 2_a, 2_b, 3_a, 3_b, 4_a, 4_b, 5_a, 7_a, 7_b$. We zien dat gelijke waarden in de uitvoer in dezelfde volgorde ten opzichte van elkaar staan als ze in de invoer stonden.

b. Het algoritme sorteert nog wel maar is niet stabiel. Probeer het maar eens.

Opgave 46



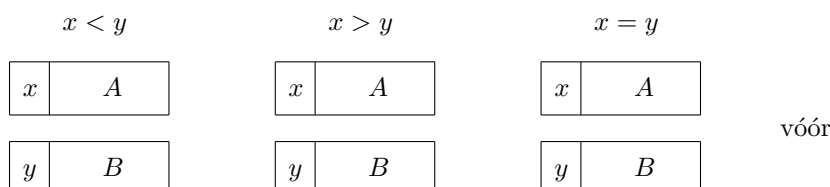
Merk op dat Radix sort ook kan worden gebruikt om woorden van verschillende lengte te sorteren. In dat geval moeten we tijdelijk alle woorden even lang maken, bijvoorbeeld door ze aan te vullen met een karakter dat ‘voor’ alle andere karakters moet worden geordend.

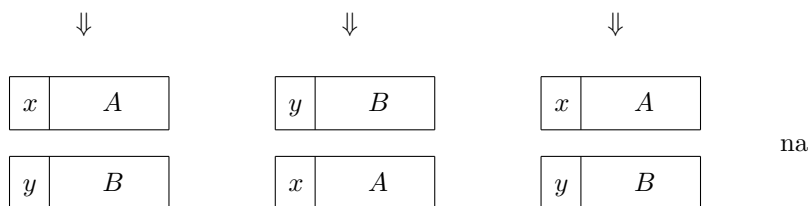
b. Aan te tonen is dat Radix sort werkt; dit doen we door middel van inductie. Hiertoe bewijzen we met behulp van inductie dat na de i -de ronde de woorden corresponderend met de laatste i karakters gesorteerd zijn. **Merk op:** het is van belang dat het algoritme dat we voor de sorteerrondes gebruiken stabiel is.

Basisgeval is $i = 1$, werkt per definitie, want we gebruiken een correct sorteer-algoritme.

Inductiehypothese: stel dat de woorden corresponderend met de laatste i karakters correct gesorteerd zijn en we nu gaan sorteren op het $i + 1$ -de karakter met een stabiel sorteer-algoritme (zoals Counting sort).

We moeten nu aantonen dat de (deel)woorden die corresponderen met de laatste $i + 1$ karakters daarna ook weer correct gesorteerd staan.





In het plaatje hiervoor bekijken we twee woorden, waarvan A en B de (deel)woorden (of getallen) voorstellen die corresponderen met de laatste i karakters, met $A \leq B$. We zien wat er vóór het sorteren op het $i + 1$ -e karakter geldt, en wat erna.

Voor de karakters op positie $i + 1$ zijn er drie mogelijkheden: $x < y$, $x > y$ of $x = y$. Als $x < y$ moet het (deel)woord dat begint met x altijd eerder in de ordening komen dan het andere (d.w.z. $xA < yB$), ongeacht wat erachter staat, en dat is ook precies wat sorteren op het $i + 1$ -karakter oplevert. Analoog het geval dat $x > y$; in dat geval is altijd $yB < xA$. Voor het geval $x = y$ geldt, omdat $A \leq B$, dat $xA \leq yB$. Omdat de sorteermethode op karakters stabiel is, blijft de volgorde van xA ten opzichte van yB hetzelfde als het was, en dus staan ze na afloop in de juiste volgorde. Hiermee is het bewijs compleet.

Opgave 47

Zie de uitwerking op de website.

Opgave 48

Zie de uitwerking op de website.

Opgave 2, tentamen 4 juni 2019

a. Om A en B , beide $n \times n$ -matrices te vermenigvuldigen splitsen we alle betrokken matrices (zie de opgave) in vier $\frac{n}{2} \times \frac{n}{2}$ matrices. We berekenen het product zoals in de opgave voorgesteld. Gevraagd wordt het aantal getalsoptellingen $P(n)$ dat hierbij wordt gedaan.

Merk op dat het vermenigvuldigen van twee $n \times n$ -matrices wordt teruggebracht tot 8 keer het vermenigvuldigen van twee $\frac{n}{2} \times \frac{n}{2}$ -matrices. Dat verklaart de term $8P(\frac{n}{2})$.

Als we de $\frac{n}{2} \times \frac{n}{2}$ -matrices vermenigvuldigd hebben moeten we nog 4 keer twee $\frac{n}{2} \times \frac{n}{2}$ -matrices optellen. Dat worden $4 \cdot \frac{n^2}{4} = n^2$ getalsvermenigvuldigingen.

Immers, een $\frac{n}{2} \times \frac{n}{2}$ -matrix heeft $\frac{n^2}{4}$ elementen.

Als $n = 1$ moeten we twee 1×1 -matrices vermenigvuldigen, wat neerkomt op twee getallen vermenigvuldigen. Er worden in dat geval geen optellingen gedaan (maar wel één vermenigvuldiging).

b. We proberen we de oplossing te vinden door $P(n)$ herhaald in zichzelf in te vullen.

$$\begin{aligned}
P(n) &= 8 \cdot P\left(\frac{n}{2}\right) + n^2 = 8 \cdot \left(8 \cdot P\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2}\right) + n^2 = 8^2 \cdot P\left(\frac{n}{2^2}\right) + 2n^2 + n^2 \\
&= 8^2 \cdot \left(8 \cdot P\left(\frac{n}{2^3}\right) + \frac{n^2}{2^4}\right) + 2n^2 + n^2 = 8^3 \cdot P\left(\frac{n}{2^3}\right) + 4n^2 + 2n^2 + n^2 = \dots = (\text{algemene vorm}) \\
&8^\ell \cdot P\left(\frac{n}{2^\ell}\right) + 2^{\ell-1}n^2 + \dots + 2^2n^2 + 2n^2 + n^2 = 8^\ell \cdot P\left(\frac{n}{2^\ell}\right) + n^2 \cdot (2^{\ell-1} + \dots + 2^2 + 2 + 1) = 8^\ell \cdot P\left(\frac{n}{2^\ell}\right) + n^2 \cdot (2^\ell - 1).
\end{aligned}$$

Gebruik nu de beginvoorwaarde $P(1) = 0$ en dat $n = 2^k$ en kies vervolgens $\ell = k$. Dan:

$$P(n) = 8^k \cdot P(1) + n^2 \cdot (2^k - 1) = n^3 \cdot 0 + n^2 \cdot (n - 1) = n^2 \cdot (n - 1).$$

c. Nu nog met behulp van volledige inductie bewijzen dat $P(n) = n^2 \cdot (n - 1)$ inderdaad de oplossing is van de recurrente betrekking.

(i) Basis: Omdat $1^2 \cdot (1 - 1) = 0$ is inderdaad $P(1) = 0$, wat klopt met de recurrente betrekking.

(ii) Inductie-aanname: stel dat $P(n)$, gedefinieerd door de recurrente betrekking, gelijk is aan $n^2 \cdot (n - 1)$ voor alle tweemachten $< N$ en N ook een tweemacht.

(iii) Vervolg bewijs: we moeten nu laten zien dat de formule uit **b** ook geldt voor $n = N$, dus dat $P(N) = N^2 \cdot (N - 1)$.

$$\begin{aligned}
\text{Er geldt: } P(N) &= (\text{recurrente betrekking}) \ 8P\left(\frac{N}{2}\right) + N^2 = (\text{inductie-aanname; } \frac{N}{2} < N) \\
&8 \cdot \frac{N^2}{2} \cdot \left(\frac{N}{2} - 1\right) + N^2 = 2N^2 \cdot \left(\frac{N}{2} - 1\right) + N^2 = N^3 - 2N^2 + N^2 = N^3 - N^2 = N^2 \cdot (N - 1).
\end{aligned}$$

QED

d. Bekijk nu het aantal getalsvermenigvuldigingen $V(n)$.

Omdat het vermenigvuldigen van twee $n \times n$ -matrices wordt teruggebracht tot 8 keer het vermenigvuldigen van twee $\frac{n}{2} \times \frac{n}{2}$ -matrices krijgen we ook hier een term $8V(\frac{n}{2})$. Verder geen vermenigvuldigingen anders dan die binnen de matrix-vermenigvuldiging zelf, dus $V(n) = 8V(\frac{n}{2})$. Voor $n = 1$ worden twee getallen vermenigvuldigd, dus $V(1) = 1$.

Derhalve:

$$V(n) = \begin{cases} 1 & n = 1 \\ 8V(\frac{n}{2}) & n \geq 2, n = 2^k \end{cases}$$

We lossen deze recurrente betrekking op via de substitutiemethode.

$$\begin{aligned}
V(n) &= 8V\left(\frac{n}{2}\right) = 8 \cdot 8V\left(\frac{n}{4}\right) = 8^2 \cdot V\left(\frac{n}{2^2}\right) = 8^2 \cdot 8V\left(\frac{n}{2^3}\right) = 8^3 \cdot V\left(\frac{n}{2^3}\right) = \dots = \\
&(\text{algemene vorm}) \ 8^\ell \cdot V\left(\frac{n}{2^\ell}\right).
\end{aligned}$$

We kiezen nu $\ell = k$ en krijgen dan: $V(n) = 8^k V(1) = 8^k = 2^{3k} = (2^k)^3 = n^3$.