

Complexiteit 2025 - Huiswerk 2

a. We zoeken het paar (p, q) waarvoor geldt dat $A[p]$ de grootste en $A[q]$ de kleinste waarde uit A is. We bekijken eerst hoeveel antwoorden er kunnen zijn.

De $\frac{n}{4}$ deelrijtjes van 4 elementen zijn allemaal oplopend gesorteerd. Dat betekent dus dat er $\frac{n}{4}$ kandidaten zijn voor de grootste, namelijk van elk deelrijtje het laatste element. Als de grootste waarde op plek $\tilde{p}$ staat zijn er nog $\frac{n}{4}$ mogelijkheden voor de op-een-na-grootste, namelijk alle resterende laatste elementen van de deelrijtjes behalve die op plek $\tilde{p}$, plus het op een na grootste element uit het deelrijtje van de grootste (dus het array-element op positie $\tilde{p} - 1$). Alle andere elementen kunnen nooit de grootste of de op-een-na-grootste zijn. Dat betekent dus dat er in totaal $\frac{n}{4} \cdot \frac{n}{4} = \frac{n^2}{16}$ mogelijke antwoorden zijn.

Verder weten we van college dat (i) de hoogte van de beslissingsboom die correspondeert met een algoritme gelijk is aan het aantal arrayvergelijkingen dat dat algoritme doet in de worst case. Bovendien geldt (ii) dat het aantal bladeren van de beslissingsboom groter of gelijk is aan het aantal mogelijke antwoorden/uitkomsten (want elk antwoord dat kan voorkomen moet door het algoritme gevonden kunnen worden en de antwoorden staan altijd in een blad): $\# \text{ bladeren} = b \geq \# \text{ antwoorden}$. Ten slotte (iii) geldt voor binaire bomen dat $h \geq \lceil \lg b \rceil$.

Alles bij elkaar geeft dit:

het aantal arrayvergelijkingen in de worst case $= h \geq \lceil \lg b \rceil \geq \lceil \lg(\# \text{ antwoorden}) \rceil \geq \lceil \lg(\frac{n^2}{16}) \rceil = \lceil \lg n^2 - \lg 16 \rceil = \lceil 2 \lg n \rceil - 4$.

b. We hadden al gezien: alle in totaal $\frac{n}{2}$ array-elementen die op de voorste twee posities van elk gesorteerd deelrijtje staan kunnen zeker niet de grootste of de op-een-na-grootste zijn, want er zijn altijd zeker al twee elementen groter. De grootste en de op-een-na-grootste bevinden zich dus tussen de overige $\frac{n}{2}$ array-elementen: van elk deelrijtje de twee grootste waardes, dus de waardes op de posities $3 + 4i$ en $4 + 4i$ voor $i = 0, \dots, \frac{n}{4} - 1$. We kunnen dus volstaan met het toepassen van de toernooimethode op die elementen.

De toernooimethode doet op m elementen $m + \lceil \lg m \rceil - 2$ arrayvergelijkingen (zie college 3), dus op $\frac{n}{2}$ elementen worden $\frac{n}{2} + \lceil \lg \frac{n}{2} \rceil - 2 = \frac{n}{2} + \lceil \lg n - \lg 2 \rceil - 2 = \frac{n}{2} + \lceil \lg n \rceil - 3$ arrayvergelijkingen gedaan.

c. Binnen elk rijtje van 4 array-elementen zijn er geen inversies (want oplopend gesorteerd). Het maximale aantal inversies heb je als alle andere paren array-elementen $(A[i], A[j])$ (met $i < j$) wel een inversie zijn. We kunnen nu op twee manieren het maximale aantal inversies van A berekenen.

Manier 1: Het maximale aantal inversies in een willekeurig array van lengte n is $\binom{n}{2} = \frac{1}{2}n(n-1)$ (alle paren staan verkeerd om). Elk deelrijtje bevat $\binom{4}{2} = 6$ paren die allemaal juist wel goed staan en dus geen inversie zijn. Er zijn $\frac{n}{4}$ van die deelrijtjes, dus in totaal $6 \cdot \frac{n}{4} = \frac{3n}{2}$ paren array-elementen die zeker geen inversie zijn. Het maximale aantal inversies is dus $\frac{1}{2}n(n-1) - \frac{3n}{2} = \frac{1}{2}n^2 - 2n = \frac{1}{2}n(n-4)$. Merk op dat je nog wel moet laten zien dat dit aantal ook echt kan voorkomen. Dat doen we zometeen.

Manier 2: Het maximale aantal inversies komt voor als alle elementen uit een deelrijtje wel een inversie hebben met alle elementen uit de andere deelrijtjes. We tellen van links naar rechts per deelrijtje hoeveel inversies elk deelrijtje oplevert met alles wat erachter staat. We vinden zo: $4 \cdot (n-4) + 4 \cdot (n-8) + \dots + 4 \cdot 8 + 4 \cdot 4 = 4 \cdot (n-4 + n-8 + \dots + 8 + 4) = 16 \cdot (\frac{n}{4} - 1 + \frac{n}{4} - 2 + \dots + 2 + 1) = 16 \cdot \frac{1}{2} \cdot \frac{n}{4} (\frac{n}{4} - 1) = 8 \cdot \frac{n}{4} (\frac{n}{4} - 1) = 2n(\frac{n}{4} - 1) = \frac{1}{2}n(n-4)$.

Het maximale aantal inversies komt voor als alle elementen uit een deelrijtje wel een inversie hebben met alle elementen uit de andere deelrijtjes dus d.e.s.d.a. elk element in deelrijtje j groter is dan alle elementen die achter dat deelrijtje staan en kleiner dan alles wat voor dat deelrijtje staat. Dat betekent dus dat het vierde element van elk deelrijtje kleiner is dan alles wat voor het deelrijtje staat, en dat het eerste element van het deelrijtje groter is dan alles wat achter het deelrijtje staat.

Een voorbeeld voor $n = 12$: $A = 9, 10, 11, 12, 5, 6, 7, 8, 1, 2, 3, 4$. Aantal inversies = 48.

d. Bubblesort is een compare-exchange (vergelijk-en-verwissel) algoritme dat gebruikmaakt van buurverwisselingen. Dat betekent dat er per arrayvergelijking hooguit één inversie wordt opgeheven. In dat geval geldt dat het aantal arrayvergelijkingen dat gedaan moet worden ten minste gelijk is aan het aantal inversies, want een gesorteerd array heeft nul inversies. In het bijzonder geldt: het aantal arrayvergelijkingen in de *worst case* $\geq$ het maximale aantal inversies dat in A kan voorkomen $= \frac{1}{2}n(n-1)$.

e. We bepalen de vier grootste arrayelementen (van n stuks) zoals in de opgave staan aangegeven. Eerst de grootste waarde van $A[1]$ t/m $A[4]$ bepalen: dat kost 3 arrayvergelijkingen; dan de grootste van de overige drie: dat kost 2 arrayvergelijkingen; ten slotte nog 1 arrayvergelijking om de grootste en kleinste van de twee resterende elementen te vinden. Samen 6 arrayvergelijkingen.

Opmerking: eigenlijk hebben we hier de juiste ordening van de vier elementen bepaald, dus gesorteerd. Je kunt hier ook de vier arrayelementen sorteren met -bijvoorbeeld- Insertion sort. In dat geval is het *worst case* aantal arrayvergelijkingen ook 6. Sorteren van 4 waardes kan overigens in de *worst case* met 5 arrayvergelijkingen (bijvoorbeeld Mergesort).

Neem nu de volgende vier arrayelementen, bepaal daar de grootste, een-na-grootste, twee-na-grootste en drie-na-grootste waarde van (dat kost weer 6 arrayvergelijkingen) en vergelijk deze met de tot dusver gevonden grootste vier waarden om de grootste vier van de eerste acht arrayelementen (het voorstuk) te vinden. Dat kan met 4 arrayvergelijkingen. Namelijk vergelijk eerst de grootste van het eerste viertal met de grootste van het tweede viertal; dat geeft de grootste van alle acht. Stel dat die uit het eerste viertal kwam; dan vergelijk je vervolgens de grootste van het tweede viertal met de een-na-grootste van het eerste viertal om de op-een-na-grootste van alle acht te vinden, en zo iets nog twee keer. Dit komt overeen met het merge-algoritme totdat je de grootste vier waarden hebt. Vervolgens doen we herhaald hetzelfde, dus telkens de vier volgende array-elementen erbij nemen; per stap (dus toegevoegd viertal) kost dit 10 arrayvergelijkingen en we moeten $\frac{n}{4} - 1$ stappen doen. In totaal komen we zo op: $6 + (\frac{n}{4} - 1) \cdot 10 = \frac{5}{2}n - 4$ arrayvergelijkingen.

Als je doorrekent met telkens 5 arrayvergelijkingen (in de *worst case*) om de volgende vier array-elementen te sorteren, kom je op telkens maximaal 9 arrayvergelijkingen per nieuw viertal, hetgeen $5 + (\frac{n}{4} - 1) \cdot 9 = \frac{9}{4}n - 4$ arrayvergelijkingen in de *worst case* oplevert¹.

f. Merk allereerst op dat: $T(n) = T(n-4) + \frac{5}{2}n - 4$, dus $T(n-4) = T(n-8) + \frac{5}{2}(n-4) - 4$ en $T(n-8) = T(n-12) + \frac{5}{2}(n-8) - 4$.

Vul nu de recurrente betrekking herhaald in zichzelf in.

$$\begin{aligned} T(n) &= T(n-4) + \frac{5}{2}n - 4 = T(n-8) + \frac{5}{2}(n-4) - 4 + \frac{5}{2}n - 4 = T(n-8) + \frac{5}{2} \cdot (n+n-4) - 2 \cdot 4 = \\ &= T(n-12) + \frac{5}{2}(n-8) - 4 + \frac{5}{2} \cdot (n+n-4) - 2 \cdot 4 = T(n-16) + \frac{5}{2} \cdot (n+n-4+n-8) - 3 \cdot 4 = \dots = \end{aligned}$$

¹Als je dat gedaan hebt wordt het ook goedgerekend.

(vermoeden van de algemene vorm) $T(n - \ell \cdot 4) + \frac{5}{2} \cdot (n + n - 4 + \dots + n - \ell \cdot 4 + 4) - \ell \cdot 4$. Dit geldt dan voor elke ℓ .

We weten dat $T(0) = 0$, dus we kiezen nu ℓ zo dat $n - \ell \cdot 4 = 0$, dus $\ell = \frac{n}{4}$ ($= k$, want $n = 4k$). Dan vinden we: $T(n) = T(0) + \frac{5}{2} \cdot (n + n - 4 + \dots + 4) - \frac{n}{4} \cdot 4 = \frac{5}{2} \cdot 4 \cdot (\frac{n}{4} + \frac{n}{4} - 1 + \dots + 1) - n = 10 \cdot \frac{1}{2} \cdot \frac{n}{4} \cdot (\frac{n}{4} + 1) - n = \frac{5}{16}n^2 + \frac{5}{4}n - n = \frac{5}{16}n^2 + \frac{1}{4}n$.

Nu nog met volledige inductie bewijzen dat $\frac{5}{16}n^2 + \frac{1}{4}n$ inderdaad de oplossing is van de recurrente betrekking.

(i) Basis: voor $n = 0$ geldt dat $\frac{5}{16}0^2 + \frac{1}{4}0 = 0$, wat klopt met de recurrente betrekking.

(ii) Inductie-aanname: stel dat $T(n)$, de oplossing van de recurrente betrekking, voor alle viervouden $n < N$, met N ook een viervoud, gelijk is aan $\frac{5}{16}n^2 + \frac{1}{4}n$.

(iii) We moeten nu laten zien dat onder deze aanname de formule ook geldt voor N , dus dat $T(N) = \frac{5}{16}N^2 + \frac{1}{4}N$.

$T(N) =$ (recurrente betrekking) $T(N - 4) + \frac{5}{2}N - 4 =$ (inductie-aanname; $N - 4 < N$) $\frac{5}{16}(N - 4)^2 + \frac{1}{4}(N - 4) + \frac{5}{2}N - 4 = \frac{5}{16}(N^2 - 8N + 16) + \frac{1}{4} \cdot N - \frac{1}{4} \cdot 4 + \frac{5}{2}N - 4 = \frac{5}{16}N^2 - \frac{5}{2}N + 5 + \frac{1}{4}N - 1 + \frac{5}{2}N - 4 =$ (er valt nu van alles weg) $\frac{5}{16}N^2 + \frac{1}{4}N$. Q.E.D.