(Beknopte) antwoorden opgaven werkcollege 3

Opgave 19

a. We kunnen in dit geval het aantal keer dat de arrayvergelijkingen gedaan worden exact berekenen, wat de opgave wat makkelijker maakt. We zien dat elk van de drie while-loops een vast aantal iteraties doet, namelijk alle drie $\frac{n}{2}$ keer (want n is even). De arrayvergelijkingen in regels 3, 10 en 17 gebeuren dus ook elk precies $\frac{n}{2}$ keer $\Rightarrow \Theta(n)$. Nu de andere regels daarmee in verband brengen.

- Regels 1, 8 en 15 gebeuren elk één keer $\Rightarrow \Theta(1) \subset O(n)$.
- Regels 2, 8 en 16 gebeuren elk $\frac{n}{2}+1$ keer (één keer vaker dan 3, 10 en 17) $\Rightarrow \Theta(n) \subseteq O(n)$.
- Regels 4, 11 en 18 gebeuren elk hooguit even vaak als resp. regels 3, 10 en 17 $\Rightarrow O(n)$.
- Regels 6, 13 en 20 gebeuren elk precies even vaak als resp. regels 3, 10 en 17 $\Rightarrow O(n)$.

De arrayvergelijkingen zijn zinnige, fundamentele operaties, en vanwege bovenstaande, samen ook maatgevend voor de complexiteit. In totaal doet het algoritme er $\frac{3}{2}n$.

b. We tellen hier het aantal toekenningen waarbij array-elementen betrokken zijn, dus geen teller-toekenningen zoals $i := 2$. Merk verder op dat na de eerste while-loop telkens de grootste waarde van $A[i]$ en $A[i+1]$ (i oneven) op de even positie staat en de kleinste van de twee op de oneven plek.

- Regel 4 gebeurt hooguit $\frac{n}{2}$ keer, en dat komt voor d.e.s.d.a. $A[i] > A[i+1]$ voor *alle* oneven $1 \leq i \leq n$. Regel 4 wordt minstens 0 keer uitgevoerd, en dat laatste gebeurt d.e.s.d.a. $A[i] \leq A[i+1]$ voor alle oneven $1 \leq i \leq n$. Dat zijn dus minimaal 0 en maximaal $\frac{3}{2}n$ (functie $Wissel$) toekenningen.
- Regels 8 en 15 gebeuren elk altijd één keer. Samen dus altijd 2 toekenningen.
- Regel 11 gebeurt 0 keer d.e.s.d.a. $\min(A[1], A[2])$ het (of eigenlijk een) kleinste element is van A. Hij gebeurt maximaal $\frac{n}{2} - 1$ (namelijk niet voor $n = 1$) keer als telkens een nieuw minimum wordt gevonden, d.w.z. d.e.s.d.a. $\min(A[i], A[i+1]) < \min(A[i-2], A[i-1])$ voor alle oneven $3 \leq i \leq n$.
- Regel 18 is analoog aan regel 11: minimaal 0 keer d.e.s.d.a. $\max(A[1], A[2])$ het grootste element is van A, en maximaal $\frac{n}{2} - 1$ keer als $\max(A[i], A[i+1]) > \max(A[i-2], A[i-1])$ voor alle oneven $3 \leq i \leq n$.

Samengevat:

- Best case: 2 toekenningen, alleen als alle paren al goed staan voor het wisselen in regel 4 en als $A[1] \leq A[i]$ voor alle $1 \leq i \leq n$.
- Worst case: $\frac{5}{2}n$, alleen als alle paren verkeerd om staan voor het wisselen in regel 4 en als de deelrijtjes (na het wisselen) op de even en oneven posities resp. aflopend en oplopend zijn gesorteerd.

c. Er zijn meerdere manieren om het algoritme aan te passen. Een drietal conceptueel eenvoudige, in geval van oneven n:

1. Dupliceer het laatste element zodat je een even aantal elementen krijgt (namelijk $n+1$) en voer het algoritme daarop uit. Het aantal vergelijkingen is dan $3\lceil \frac{n}{2} \rceil$.
2. Voer het algoritme uit op de eerste $n-1$ elementen en vergelijk op het eind de waarde $A[n]$ met de gevonden min en max. Dat is maximaal $3\lfloor \frac{n}{2} \rfloor + 2$ vergelijkingen voor n oneven (en maximaal $\frac{3}{2}n$ voor n even).
3. Tussen regels 7 en 8: als $A[n] > A[n-1]$, verwissel de twee. Vervolgens mag je $A[n]$ negeren (dit is dan immers gegarandeerd geen maximum en geen minimum) en voer je de rest van het algoritme uit op de eerste $n-1$ elementen. Het aantal vergelijkingen is dan $3\lfloor \frac{n}{2} \rfloor + 1$ voor n oneven.

Opgave 20

Zie uitwerking website

Opgave 22

a. Een triviaal algoritme vergelijkt alle (ongeordende) paren in een of andere volgorde, bijvoorbeeld $(A[1], A[2]), (A[1], A[3]), \ldots, (A[1], A[n]), (A[2], A[3])$, $(A[2], A[4]), \ldots, (A[2], A[n]), (A[3], A[4]), \ldots$ en stopt zodra het twee elementen tegenkomt die niet gelijk zijn. Het geeft True d.e.s.d. als alle elementen verschillend zijn. Dit algoritme doet 1 vergelijking in de best case (het vindt direct twee dezelfde) en $\frac{1}{2}n(n-1)$ vergelijkingen in de worst case ((i) alle bekeken paren zijn verschillend, of (ii) alle verschillend, maar het laatst gecontroleerde paar elementen heeft dezelfde waarde).

b. Ad hoc: analoog aan 3e (zie uitwerking website), uit het ongerijmde.
Stel er bestaat een *correct* algoritme dat in de worst case (strikt) minder dan $\frac{1}{2}n(n-1)$ vergelijkingen doet. Dan doet dat algoritme voor *alle* mogelijke invoer echt minder dan $\frac{1}{2}n(n-1)$ vergelijkingen, dus in het bijzonder voor rijtjes waarin alle elementen verschillend zijn. Neem een rijtje B met allemaal verschillende elementen en laat het algoritme daarop los. Het algoritme geeft dan als antwoord True voor invoer B en doet doet daarop echt minder dan $\frac{1}{2}n(n-1)$ vergelijkingen. Dan is er dus ten minste één paar $B[i], B[j]$ dat niet wordt vergeleken. We construeren nu een invoerrijtje C:

$$C[k] = \begin{cases} B[k] & k \neq j \\ B[i] & k = j \end{cases}$$

Alle elementen van C zijn dus verschillend, op $C[i]$ en $C[j]$ na. We voeren het algoritme nu uit op invoer C. Het algoritme is deterministisch en alle gedane vergelijkingen leveren derhalve hetzelfde resultaat op als voor B, en zal dus ook de elemnten op positie i en j niet vergelijken. Het algoritme geeft derhalve voor C ook de uitvoer True, wat in dit geval het verkeerde antwoord is. Tegenspraak met de correctheid van het algoritme.

Merk op dat dit betekent dat het algoritme uit **a.** optimaal is in zijn klasse.

Toegift: met een adversary argument: geef telkens aan dat twee elementen verschillend zijn. Per vergelijking kan je dan precies één (ongeordend) paar afstrepen, en je hebt er $\frac{1}{2}n(n-1)$.

c. Als we ook vergelijkingen van de vorm $A[i] < A[j]$ mogen doen kunnen we een sneller algoritme geven. Eerst sorteren, waarbij eventueel gelijke waarden naast elkaar komen te staan. We hoeven dan alleen nog buren te vergelijken. Sorteren kan in $\Theta(n \lg n)$ en dan hoef je alleen nog $n-1$ buurparen te vergelijken.

Opgave 27

a. Zie uitwerking website

b. Ter herinnering: het aantal bladeren van de beslissingsboom die met je algoritme correspondeert moet altijd minstens zo groot zijn als het aantal mogelijke antwoorden, omdat elk antwoord door je algoritme gevonden moet kunnen worden. Zie college 3.
Opties voor de vijf grootste waarden (ongeordend): $\binom{n}{5}$, dus $b \geq \binom{n}{5} = \frac{n(n-1)(n-2)(n-3)(n-4)}{120}$. Dan geldt: het aantal vergelijkingen in de worst case = hoogte van de met het algoritme corresponderende beslissingsboom = $h \geq \lceil \lg b \rceil \geq \lceil \lg \frac{n(n-1)(n-2)(n-3)(n-4)}{120} \rceil = \lceil \lg(n(n-1)(n-2)(n-3)(n-4)) - \lg 120 \rceil = \lceil \lg n + \lg(n-1) + \lg(n-2) + \lg(n-3) + \lg(n-4) - \lg 120 \rceil \in \Theta(\lg n)$.

c. We mogen weer aannemen dat A en B verschillende waarden bevatten. Het aantal manieren om de twee gesorteerde rijtjes A en B samen te voegen tot één gesorteerde rij is $\binom{n+m}{n}$. Immers, het resulterende rijtje na samenvoegen is $n+m$ lang, en wordt uniek bepaald door de posities van de n elementen van A te geven. (Alternatief: door de posities van de m elementen van B te geven; dat levert hetzelfde antwoord op, want $\binom{n+m}{n} = \binom{n+m}{m}$.)

Dan geldt: aantal vergelijkingen in de worst case $= h \geq \lceil \lg b \rceil \geq \lceil \lg \binom{n+m}{n} \rceil$. Dat verder uitwerken is naar, maar niet nodig. Misschien interessant:

- Als $m = n$ geldt: $\lceil \lg \binom{n+m}{n} \rceil = \lceil \lg \binom{2n}{n} \rceil = \lceil \lg(2n)! - 2 \lg n! \rceil \approx 2n \lg(2n) - 2n \lg n = 2n + 2n \lg n - 2n \lg n = 2n$.
- $\lceil \lg \binom{n+m}{n} \rceil = \lceil \lg n + \lg(n-1) + \ldots + \lg(n-m) \rceil \leq \lceil m \lg n \rceil$, dus $h \in O(m \lg n)$
- $\lceil \lg \binom{n+m}{n} \rceil = \lceil \lg n + \lg(n-1) + \ldots + \lg(n-m) \rceil \geq \lceil m \lg(n-m) \rceil$, dus $h \in \Omega(m \lg(n-m))$
- preciezere afschatting: zie `https://math.stackexchange.com/questions/64716/approximating-the-logarithm-of-the-binomial-coefficient`

Opgave 23, 24

Zie uitwerking website