Beknopte antwoorden opgaven werkcollege 5

Opgave 15

a.

- als $n = 2$ volstaat één vergelijking voor het bepalen van de grootste en op één na grootste.
- als $n \geq 4$ hebben we twee recursieve aanroepen op een rijtje ter grootte $\frac{n}{2}$. Vervolgens kan je met één vergelijking de grootste van het hele rijtje bepalen (nml. de grootste van de twee grootsten), en met nog één de op één na grootste (nml. de kleinste van de eerste vergelijking met de op één na grootste van de helft met de grootste). Dit komt samen neer op $2G(\frac{n}{2}) + 2$.

b. $\frac{3}{2}n - 2$

Opgave 16

Zie uitwerking website

Opgave 31

Zie uitwerking website

Opgave 23

Zie uitwerking website

Opgave 24

Zie uitwerking website

Opgave 18

a.

- vergelijken is een zinnige operatie voor het vinden van een major
- de test op regel 12 wordt $n - i$ keer uitgevoerd (mits $i \leq \lceil \frac{n}{2} \rceil$ en $\mathit{found} = \mathit{False}$ en $\mathit{gehad}[i] = \mathit{False}$), dus gezien i begint op 1 en *found* en $\mathit{gehad}[i]$ op *False*, minstens $n - 1 \in \Omega(n)$ keer.
- regels 1, 2, 3, 5 en 6 gebeuren elk respectievelijk 1, $n + 1$ (of 1), n, 1 en 1 keer; allemaal in $\Omega(n)$.
- de tests op regel 7 gebeuren hooguit $\lceil \frac{n}{2} \rceil + 1$ keer; regels 8 en 21 hooguit even vaak als regel 7; regels 9, 10 en 17 hooguit even vaak als regel 8; regel 18 hooguit even vaak als regel 17; daarmee allemaal in $\Omega(n)$.
- regel 13 gebeurt hooguit even vaak als regel 12; regel 15 even vaak als regel 12.

- regel 11 gebeurt hooguit $\lceil \frac{n}{2} \rceil$ keer vaker dan regel 12; daarmee hooguit een constante factor vaker (want $\Omega(n)$), en daarmee (hooguit) in dezelfde orde van grootte.

b.

- $i = 1$: *gehad*[1], *gehad*[3] en *gehad*[7] worden *True*; niet gevonden ($3 < 5\frac{1}{2}$), door; $n - 1 = 10$ vergelijkingen
- $i = 2$: *gehad*[2] en *gehad*[4] worden *True*; niet gevonden ($2 < 5\frac{1}{2}$), door; $n - 2 = 9$ vergelijkingen
- $i = 3$: *gehad*[3] $=$ *True*, sla over; 0 vergelijkingen
- $i = 4$: *gehad*[4] $=$ *True*, sla over; 0 vergelijkingen
- $i = 5$: *gehad* wordt *True* op posities $5, 6, 8, 9, 10, 11$; gevonden ($6 > 5\frac{1}{2}$), stop; $n - 5 = 6$ vergelijkingen

Totaal $10 + 9 + 6 = 25$ vergelijkingen.

c. in het slechtste geval $\sum_{i=1}^{\lceil \frac{n}{2} \rceil}(n - i)$ vergelijkingen, als je $\lceil \frac{n}{2} \rceil$ keer op regel 9 t/m 19 komt; de eerste $\lceil \frac{n}{2} \rceil$ elementen moeten verschillend zijn (anders sla je daar wat over vanwege *gehad*) en de eerste $\lceil \frac{n}{2} \rceil - 1$ moeten geen major zijn (anders stop je, maar in de laatste iteratie maakt dat niet uit)

d. in het beste geval $n - 1$ vergelijkingen, als je maar één keer op regel 9 t/m 19 komt; je kan ofwel meteen een major vinden (dan wordt *found True*), of de eerste $\lceil \frac{n}{2} \rceil$ elementen moeten hetzelfde zijn (wat bij even n geen major hoeft te betekenen) zodat je elke keer *gehad*[i] op *True* hebt.

e. in het slechtste geval doe je nog steeds $\sum_{i=1}^{\lceil \frac{n}{2} \rceil} i$ vergelijkingen, maar nu moeten de eerste $\lceil \frac{n}{2} \rceil - 1$ elementen uniek zijn in het array (en dus ook niet voorkomen in de tweede helft), anders sla je daar dingen over; het laatste bekeken element ($\lceil \frac{n}{2} \rceil$) mag wel voorkomen in de tweede helft, daarna stopt het algoritme toch.

Met de aanpassing is de test uit regel 12 niet meer maatgevend. Er is een klasse van invoerrijtjes te construeren waarvoor regel 11 en de nieuwe regel in orde van grootte strikt vaker worden uitgevoerd dan regel 12. Bijvoorbeeld: laat de eerste $\sqrt{n}$ elementen verschillend zijn, en laat alle andere elementen gelijk zijn aan $A[1]$ of $A[2]$, op een manier dat A geen major bevat. Dan wordt regel 12 $O(n) + O(\sqrt{n}^2) = O(n)$ keer uitgevoerd en regels 11 en de nieuwe $O(n\sqrt{n})$.