

Complexiteit

Uitwerking Opgave 6

- a) In het slechtste geval wordt de buitenste loop n (of $n - 1$; in de n -de iteratie wordt immers nooit vergeleken, dus die maakt niet uit) keer uitgevoerd, en worden daarbij iedere keer *Boven* vergelijkingen uitgevoerd, waarbij *Boven* in de eerste iteratie op $n - 1$ wordt gezet, in de volgende op $n - 2$, etcetera. *Boven* loopt dus eigenlijk van $n - 1$ tot 0, waarmee we in totaal dus $0 + 1 + \dots + n - 1 = \frac{1}{2}n(n - 1)$ vergelijkingen doen.

Om deze worst-case van het algoritme gedaan te krijgen moet de binnenste loop dus $n - 1$ keer worden gestart. Daartoe moet na elk van de eerste $n - 2$ iteraties de boolean *Gewisseld* op True zijn gezet, m.a.w. er moet in elk van de eerste $n - 2$ iteraties gewisseld worden; het maakt niet uit of er in de laatste iteratie gewisseld wordt, de vergelijking gebeurt altijd.

Elk element dat te veel naar rechts staat (te klein is) wordt, per iteratie, één plaats naar links verschoven. We hebben dus de eerste $n - 2$ iteraties ten minste één wissel precies als er een element is dat ten minste $n - 2$ plaatsen naar links verschoven moet worden (als alle elementen minder verschoven moeten worden stoppen we eerder, omdat dan in een eerdere iteratie alle elementen al op hun plaats staan). Dit kan alleen maar de waarde op plek n en/of op plek $n - 1$ zijn. In de worst case moet dus ofwel $A[n - 1]$ de kleinste zijn van $A[1], \dots, A[n - 1]$, of moet $A[n]$ de kleinste of één-na kleinste van de hele array zijn.

- b) Aangezien *Gewisseld* op True wordt geïnitieerd zal de binnenste for-loop ten minste één keer worden uitgevoerd, dus de vergelijking $A[j] > A[j + 1]$ gebeurt ten minste $n - 1$ keer. In het beste geval worden ook precies alleen die $n - 1$ vergelijkingen gedaan, en dat komt voor als meteen na de eerste ronde *Gewisseld* False is. Dat is het geval dan en slechts dan als voor alle $j = 1, 2, \dots, n - 1$ $A[j] \leq A[j + 1]$. En dat betekent precies dat $A[1] \leq A[2] \leq \dots \leq A[n]$, ofwel dat het array olopend gesorteerd is.
- c) Het aantal arrayvergelijkingen is een goede basisoperatie. De operaties in regel 6 gebeuren hooguit even vaak als de vergelijking in regel 5. Die vergelijking wordt ten minste $n - 1$ keer uitgevoerd (eerste ronde) en regels 3 en 4 elk maximaal $n - 1$ keer, dus hooguit even vaak. De test in regel 2 gebeurt maximaal $n + 1$ keer, in orde van grootte dus hooguit even vaak als de vergelijking $A[j] > A[j + 1]$. (We nemen voor het gemak aan dat $n > 1$; voor $n = 1$ is de arrayvergelijking niet maatgevend.)
- d) Met deze aanpassing verandert de worst case niet: met een omgekeerd gesorteerd rijtje geldt dat $Onder = 1$ en $Boven = n - i$ na regel 3 voor ieder van de $n - 1$ relevante iteraties. De complexiteit blijft dus hetzelfde. Hooguit wordt het aantal worst-case invoeren beperkt.

De best case verandert ook niet; in de aangepaste versie van het algoritme wordt de buitenste loop sowieso 1 keer doorlopen, en in de best case blijft dat 1 keer. Er wordt in dat geval niet gewisseld en *Onder* en *Boven* hebben in het geheel geen effect. Het aantal vergelijkingen is dus nog steeds $n - 1$ en ook de best case invoer verandert niet.