

Uitwerking opgave 43

Te bewijzen: Shellsort met de gegeven stapgroottes/sprongafstanden doet minstens cn^2 vergelijkingen (voor zekere c) in de worst case. Het is dus voldoende om een invoer (bad case) te geven en daarvoor te laten zien dat Shellsort met de gegeven stapgroottes daarop ten minste een kwadratisch aantal vergelijkingen doet. Het bewijs gaat analoog aan het bewijs van het college waarbij Shell's stapgrootteserie werd gehanteerd.

Neem als beginrijtje een rijtje van n verschillende getallen waarbij de kleinste $\frac{n}{5}$ getallen in oplopende volgorde op de posities $1, 6, 11, \dots, 5i+1, \dots, n-4$ staan ($i = 0, 1, \dots, 5(\frac{n}{5}-1)$). We mogen wel een rijtje nemen dat bestaat uit de getallen 1 t/m n . In dat geval ziet het array er zo uit:

$$1, -, -, -, -, 2, -, -, -, -, 3, -, -, \dots, -, \frac{n}{5}, -, -, -, -$$

Dankzij de keuze van de stapgroottes (altijd 5-machten, dus deelbaar door 5) worden de elementen op posities $1, 6, 11, 16, \dots, n-4$ alleen onderling vergeleken en blijven deze getallen tot aan de laatste ronde (met stapgrootte 1) op hun plaats staan omdat ze reeds gesorteerd zijn. Er worden in die rondes wel vergelijkingen gedaan, en mogelijk ook elementen op andere posities onderling verwisseld, maar de getallen 1 t/m $\frac{n}{5}$ blijven staan. Pas in de laatste doorgang worden ze verplaatst, en op hun uiteindelijke positie gezet via Insertion sort. We hebben nu:

arrayvergelijkingen totaal $\geq$ # arrayvergelijkingen van de laatste ronde $\geq$ # arrayvergelijkingen nodig om in de laatste ronde de getallen 1 t/m $\frac{n}{5}$ goed te zetten.

Het goedzetten (via Insertion sort) van deze $\frac{n}{5}$ (waarvan de 1 al goed staat, dus er resteren er nog $\frac{n}{5} - 1$) getallen kost:

$$5 + 9 + 13 + \dots \geq 1 + 2 + 3 + \dots + \frac{n}{5} - 1 = \frac{1}{2} \cdot \frac{n}{5} \cdot (\frac{n}{5} - 1) \in \Theta(n^2)$$

arrayvergelijkingen, waarmee we het gevraagde bewezen hebben.

Overigens kost het goedzetten van $\frac{n}{5}$ hier $\frac{4n}{5} - 3$ vergelijkingen, maar dat hoeven we niet te berekenen omdat we een (nogal grove) afschatting kunnen gebruiken.

Een ander, vrijwel hetzelfde, voorbeeld van een bad case invoer zou het rijtje kunnen zijn waar de kleinste $\frac{n}{5}$ elementen in oplopende volgorde op de posities $5, 10, 15, \dots$ staan. (Misschien is dat rijtje makkelijker te onthouden?). Het array ziet er dan aan het begin zo uit:

$$-, -, -, -, 1, -, -, -, -, 2, -, -, -, \dots, -, \frac{n}{5} - 1, -, -, -, -, \frac{n}{5}$$

Ook nu is het zo dat deze $\frac{n}{5}$ getallen tot de laatste ronde (met stapgrootte 1) op hun plaats blijven staan omdat ze reeds oplopend gesorteerd zijn. Pas als we gaan 1-sorteren worden ze op hun definitieve plaats gezet (met Insertion sort). Het getal op positie 5 moet naar plek 1, het getal op plek 10 naar 2, het getal op plek 15 naar 3, $\dots$, het getal op plek $5i$ naar plek i , etcetera. Dit kost in totaal (voor de $\frac{n}{5}$ kleinste getallen dus)

$$4+9+13+17+\dots+(4i+1)+\dots+(\frac{4n}{5}+1) \geq 1+2+3+4+\dots+\frac{n}{5} = \frac{1}{2} \cdot \frac{n}{5} \cdot (\frac{n}{5}+1) \in \Theta(n^2)$$

arrayvergelijkingen.