

## (Beknpte) antwoorden opgaven werkcollege 6

### 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 kun je met één vergelijking de grootste van het hele rijtje bepalen (dat is namelijk de grootste van de twee grootsten), en met nog één de op één na grootste (dat is namelijk de kleinste van de eerste vergelijking of de op één na grootste uit de helft met de grootste). Dit komt samen neer op  $2G(\frac{n}{2}) + 2$ .

b. We vullen de uitdrukking herhaald in zichzelf in:

$$\begin{aligned} G(n) &= 2G(\frac{n}{2}) + 2 = 2 \cdot (2G(\frac{n}{2^2}) + 2) + 2 = \\ &2^2 \cdot G(\frac{n}{2^2}) + 2^2 + 2^1 = 2^2 \cdot (2G(\frac{n}{2^3}) + 2) + 2^2 + 2^1 = \\ &2^3 \cdot G(\frac{n}{2^3}) + 2^3 + 2^2 + 2^1 = \dots = (\text{vermoeden}) 2^\ell \cdot G(\frac{n}{2^\ell}) + 2^\ell + \dots + 2^3 + 2^2 + 2^1 \\ &= 2^\ell \cdot G(\frac{n}{2^\ell}) + 2^{\ell+1} - 2 = \end{aligned}$$

Gebruik nu dat  $G(2) = 1$  en dat  $n = 2^k$  en kies  $\ell = k - 1$ , dan krijgen we:  
 $= 2^{k-1} \cdot G(2) + 2^k - 2 = \frac{n}{2} + n - 2 = \frac{3}{2}n - 2$ .

Vervolgens bewijzen we door middel van inductie dat  $G(n) = \frac{3}{2}n - 2$  voor alle tweemachten.

**Basisgeval:** Voor  $n = 2$  hebben we  $\frac{3}{2} \cdot 2 - 2 = 3 - 2 = 1$ , en dat moest ook volgens de recurrente betrekking. Dus het basisgeval is afgedekt.

**Inductieaanname:** neem nu aan dat de gelijkheid geldt voor alle tweemachten  $n < N$ , met  $N$  ook een tweemacht. Dan gaan we bewijzen dat de gelijkheid ook voor  $N$  geldt, dus dat  $G(N)$ , zoals gedefinieerd door de recurrente betrekking, gelijk is aan  $\frac{3}{2}N - 2$ .

We gebruiken hieronder eerst de recurrente betrekking, dan de inductieaanname (want  $\frac{N}{2}$  is een tweemacht  $< N$ ), en vervolgens is het een kwestie van verder uitschrijven.

**Bewijs voor  $N$ :**  $G(N) = 2G(\frac{N}{2}) + 2 = 2 \cdot (\frac{3}{2} \cdot \frac{N}{2} - 2) + 2 = \frac{3}{2}N - 4 + 2 = \frac{3}{2}N - 2$ . Hiermee is het bewijs geleverd.

### Opgave 16

Zie uitwerking website

### Opgave 2 tentamen juni 2024

Zie uitwerking website

### Opgave 34

Het zo ontstane algoritme heet binary insertion sort en is zeer efficiënt wat betreft het aantal vergelijkingen.

Als we binair zoeken in een rijtje van  $n$  getallen kost dat in de worst case hooguit  $\lceil \lg(n+1) \rceil$  vergelijkingen<sup>1</sup>. In Insertion sort moeten we bij iteratie  $i$  de juiste plek opzoeken (voor het in te voegen element  $A[i]$ ) in een rijtje van  $i-1$  elementen, te beginnen met een rijtje van 1 (als  $i=2$ ) element en eindigend met een rijtje van  $n-1$  (als  $i=n$ ) elementen. Zoeken in een rijtje van  $i-1$  elementen is maximaal  $\lceil \lg(i) \rceil$  vergelijkingen. In de worst case doen we zo dus

$$\sum_{i=2}^n \lceil \lg(i) \rceil = \sum_{i=1}^n \lceil \lg(i) \rceil \geq \lceil \sum_{i=1}^n \lg i \rceil = \lceil \lg n! \rceil$$

vergelijkingen, de theoretische ondergrens voor het aantal arrayvergelijkingen nodig om de juiste sortering te vinden.

Overigens hebben we ook:  $\sum_{i=2}^n \lceil \lg(i) \rceil \leq (n-1) \lceil \lg n \rceil$ , dus het aantal arrayvergelijkingen in de worst case is in  $\Theta(n \lg n)^2$ .

Voor  $n=6$  komt het exacte aantal vergelijkingen uit op  $1+2+2+3+3=11$ . Voor  $n=7$  komen daar nog 3 vergelijkingen bij en kom je uit op 14 vergelijkingen.

Als de positie waar een waarde  $A[j]$  moet worden tussengevoegd is gevonden, moeten het arrayelement dat op die positie staat en alle elementen rechts daarvan nog naar rechts schuiven. Iets preciezer: in iteratie  $i$  (de eerste  $i-1$  elementen staan dan al in oplopende volgorde) wordt het element dat op positie  $i$  staat ingevoegd op de juiste plek ( $i=2, \dots, n$ ). Dit kan gedaan worden door elementen naar rechts te schuiven (of door buursverwisselingen te doen). In het ergste geval moet bij iedere iteratie op positie 1 worden ingevoegd; dan moeten we dus  $i-1$  elementen naar rechts schuiven. In totaal komt dit dan uit op  $1+2+3+\dots+n-1 = \sum_{i=1}^{n-1} i = \frac{1}{2}n(n-1) \in \Theta(n^2)$  verschuivingen. Het is dus niet helemaal juist om in geval van binary insertion sort arrayvergelijkingen als basisoperatie te nemen: het algoritme is weliswaar gebaseerd op deze vergelijkingen, maar het aantal vergelijkingen is niet maatgevend voor de complexiteit van het algoritme. In de worst case is er een andere (ook nog eens fundamentele) operatie die meer werk doet, namelijk verschuivingen. Deze zijn wel maatgevend voor de complexiteit. Het probleem zou kunnen worden opgeheven door een pointerlijst te gebruiken, maar dan is binair zoeken weer lastig.

Merk op dat het aantal vergelijkingen dat gedaan wordt *om de juiste sortering te vinden* met dit algoritme wel  $\Theta(n \lg n)$  is; het extra werk komt van het daadwerkelijk op de juiste plaats zetten van de array-elementen.

## Opgave 36 (voor thuis)

We vullen de uitdrukking herhaald in zichzelf in:

$$\begin{aligned} B(n) &= 2B\left(\frac{n}{2}\right) + \frac{n}{2} = 2\left(2B\left(\frac{n}{2^2}\right) + \frac{n}{2^2}\right) + \frac{n}{2} = 2^2B\left(\frac{n}{2^2}\right) + \frac{n}{2} + \frac{n}{2} = 2^2B\left(\frac{n}{2^2}\right) + 2 \cdot \frac{n}{2} = \\ &= 2^2\left(2B\left(\frac{n}{2^3}\right) + \frac{n}{2^3}\right) + 2 \cdot \frac{n}{2} = 2^3B\left(\frac{n}{2^3}\right) + \frac{n}{2} + 2 \cdot \frac{n}{2} = 2^3B\left(\frac{n}{2^3}\right) + 3 \cdot \frac{n}{2} = \dots = \\ &= (\text{vermoeden algemene vorm}) \quad 2^\ell B\left(\frac{n}{2^\ell}\right) + \ell \cdot \frac{n}{2}. \end{aligned}$$

Gebruik nu de beginconditie  $B(1) = 0$  en  $n = 2^k$ , dus neem  $\ell = k$ . Dan:

$$2^\ell B\left(\frac{n}{2^\ell}\right) + \ell \cdot \frac{n}{2} = 2^k B\left(\frac{n}{2^k}\right) + k \cdot \frac{n}{2} = 2^k B(1) + k \cdot \frac{n}{2} = k \cdot \frac{n}{2} = \frac{n}{2} \lg n.$$

<sup>1</sup>In dit geval zijn het arrayvergelijkingen, omdat we in ronde  $i$  zoeken naar de positie van  $A[i]$ .  
<sup>2</sup>want  $\lceil \lg n! \rceil \in \Theta(n \lg n)$

Vervolgens bewijzen we door middel van inductie dat  $B(n)$ , zoals gedefinieerd door de recurrente betrekking, gelijk is aan  $\frac{n}{2} \lg n$  voor alle tweemachten.

**Basisgeval:** Voor  $n = 1$  hebben we  $\frac{1}{2} \lg 1 = \frac{1}{2} \cdot 0 = 0$ , en dat moest ook volgens de recurrente betrekking. Dus het basisgeval is afgedekt.

**Inductieaanname:** neem nu aan dat  $B(n) = \frac{n}{2} \lg n$  voor alle tweemachten  $n < N$ , met  $N$  ook een tweemacht. Dan gaan we bewijzen dat de gelijkheid eveneens voor  $N$  geldt, dus dat  $B(N)$ , zoals gedefinieerd door de recurrente betrekking, gelijk is aan  $\frac{N}{2} \lg N$ .

We gebruiken hieronder eerst de recurrente betrekking, dan de inductieaanname (want  $\frac{N}{2}$  is een tweemacht  $< N$ )<sup>3</sup>, en vervolgens is het weer een kwestie van verder uitschrijven.

**Bewijs voor  $N$ :**  $B(N) = 2B(\frac{N}{2}) + \frac{N}{2} = 2 \cdot \frac{N}{2^2} \lg \frac{N}{2} + \frac{N}{2} = \frac{N}{2} (\lg N - \lg 2) + \frac{N}{2} = \frac{N}{2} \lg N - \frac{N}{2} + \frac{N}{2} = \frac{N}{2} \lg N$ .  
Hiermee is het bewijs geleverd.

### Opgave 35 (voor thuis)

Deze mag je zelf doen

---

<sup>3</sup>de inductieaanname zegt dat  $B(\frac{N}{2}) = \frac{1}{2} \cdot \frac{N}{2} \cdot \lg(\frac{N}{2}) = \frac{N}{2^2} \lg \frac{N}{2}$