# Opdracht 1 theoretische informatica

W. Oele

15 november 2024

## Opgave 1

Bepaal de herhalingsfrequentie $T(n)$ van:

- `for(i=n-1; i<n; i++) {...}`
- `for(i=n-1; i<n^2; i++) {...}`
- `for(i=n; i<n; i++) {...}`
- `i=0; while (i < n) {a[i] = 0}`
- `i=0; while (i < n) {a[i++] = 0}`

## Opgave 2

Bepaal de O-notatie van:

- $T(n) = 17n^3 - 13n^2 + 10n + 2000$
- $T(n) = 3^n - 13n$
- $T(n) = 20 \log_2 n + n^2$

## Opgave 3

Het volgende sorteeralgoritme (uitgedrukt in pseudocode) werkt de lijst $a[1 \ldots n]$ door, het eerste stuk $(1 \ldots i)$ is gesorteerd, het tweede stuk $(i + 1 \ldots n)$ is nog ongesorteerd. Elk element $i + 1$ uit het ongesorteerde lijstgedeelte wordt in het gesorteerde gedeelte tussengevoegd, waarna het gesorteerde gedeelte van de lijst met één element is toegenomen, ten koste van de ongesorteerde deellijst:

```
Sort(lijst a):
int i, j
     for i = 1 .. n-1
         for j = i+1 .. n
                if (a[i] < a[j])
                    verwissel(a[i],a[j])
```

Bepaal de O-notatie van de tijdcomplexiteit van deze pseudocode (de functie `verwissel` heeft een constant tijdgedrag $O(1)$).

## Opgave 4

*BubbleSort* maakt gebruik van herhaald verwisselen van buurelementen in een lijst. Een element wordt naar voren verplaatst indien het kleiner is dan het buurelement. Geef de tijdcomplexiteit in O-notatie van het *BubbleSort* algoritme. Is dit slechter dan de complexiteit van het sorteeralgoritme van de vorige opgave?

## Opgave 5

Bepaal de tijdcomplexiteit in O-notatie van een algoritme om alle permutaties van $n$ verschillende voorwerpen te genereren.

## Opgave 6

Bepaal de tijdcomplexiteit in O-notatie om de determinant van een $n \times n$ matrix te bepalen.

## Opgave 7

Een *polynoom* $f(x) = \sum_{i=0}^{n} a_i x^i$ ($a_0 \ldots a_n$ zijn coëfficiënten) wordt meestal uitgeschreven als:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_1 x^1 + a_0$$

Wij kunnen de polynoom $f(x)$ herschrijven met de *regel van Horner*:

$$f(x) = ((\ldots((a_n x + a_{n-1})x + a_{n-2})x \ldots)x + a_1)x + a_0$$

Bij de evaluatie van de polynoom $f(x)$ voor de waarde $x = c$ moeten alle termen berekend worden. Verklaar welke schrijfwijze de snelste resultaten geeft. Neem aan dat de atomaire instructietijd van een vermenigvuldiging groter is dan die van een optelling.

## Opgave 8

Het is bekend dat twee priemgetallen redelijk snel tot een product vermenigvuldigd kunnen worden. Het omgekeerde, het ontbinden van zo'n product in twee priemfactoren, kost veel meer moeite. In verband met een redelijke werkverdeling tussen docent en studenten, zijn docenten dol op dit soort problemen. Wij nemen problemen – die door de docent snel geconstrueerd kunnen worden en voor de studenten lastig op te lossen zijn – in de klasse Ideale-problemen op. Geef enkele voorbeelden uit de klasse Ideale-problemen. Is er, naast het onderwijs, nog een gebied waar Ideale-problemen toegepast kunnen worden? Geef enkele voorbeelden.

## Opgave 9

**9a** Schrijf een functie met parameter $n$ die de volgende berekening uitvoert:

$$\sum_{k=1}^{k=n} 2k - 1$$

**9b** Test je functie met verschillende waarden voor $n$.

**9c** Schrijf een efficiëntere versie van deze functie en bewijs dat de efficiëntere versie *altijd* hetzelfde antwoord oplevert als de oorspronkelijke versie.

**9d** Geef de herhalingsfrequentie en complexiteitsontwikkeling van beide functies.