

I

a. Informatica - Cyber Security  
b. Nec.

1.

$$1. \quad n = 60 \cdot 10^6$$

$$2. \quad \begin{aligned} n^2 &= 60 \cdot 10^6 \\ n &= \sqrt{60 \cdot 10^6} \end{aligned}$$

$$3. \quad \begin{aligned} n^3 &= 60 \cdot 10^6 \\ n &= \sqrt[3]{60 \cdot 10^6} \end{aligned}$$

$$4. \quad n! = 60 \cdot 10^6$$

$$n! \cdot n = 60 \cdot 10^6 \cdot n$$

$$\frac{n! \cdot n}{n!} = \frac{60 \cdot 10^6 \cdot n}{n!}$$

$$n = \frac{60 \cdot 10^6 \cdot n}{n!}$$

$$5. \quad n \ln n = 60 \cdot 10^6$$

$$\ln n = \frac{60 \cdot 10^6}{n}$$

$$n = e^{\frac{60 \cdot 10^6}{n}}$$

$$6. \quad \begin{aligned} n \lg n &= 60 \cdot 10^6 \\ \lg n &= \frac{60 \cdot 10^6}{n} \end{aligned}$$

$$n = 2^{\frac{60 \cdot 10^6}{n}}$$

$$7. \quad 2^n = 60 \cdot 10^6$$

$$n = \lg(60 \cdot 10^6)$$

$$8. \quad \sqrt[n]{n} = 60 \cdot 10^6$$

$$\sqrt[n]{n} = \frac{60 \cdot 10^6}{n}$$

$$n = \left( \frac{60 \cdot 10^6}{n} \right)^2$$

$$9. \quad n^{100} = 60 \cdot 10^6$$

$$n = \sqrt[100]{60 \cdot 10^6}$$

$$10. \quad 4^n = 60 \cdot 10^6$$

$$n = \log_4(60 \cdot 10^6)$$

2.

$$1. \quad \begin{aligned} \log(n^2) &= \log(n) + \log(n) \\ &= 2 \log(n) \end{aligned}$$

$f(n) \in O(\lg(n))$  omdat als we  $c=2$  nemen  
en  $n_0 \geq 0$  dan is  $2 \log(n) \geq 2 \log(n)$

$g(n) \in O(\lg(n))$  omdat als we  $c=1$  nemen  
en  $n_0 \geq 0$  dan is  $\log(n) \geq \log(n)$



$$g(n) \in O(f(n)) \text{ en } f(n) \in O(g(n))$$

$$2. f(n) \in O(g(n))$$

$$3. f(n) \in O(g(n)) \text{ en } g(n) \in O(f(n))$$

$$4. g(n) \in O(f(n))$$

$$5. g(n) \in O(f(n))$$

$$6. g(n) \in O(f(n))$$

$$7. g(n) \in O(f(n))$$

$$8. f(n) \in O(g(n))$$

3.

$$1. O(n)$$

$$2. O(n^5)$$

$$3. O(2^n)$$

$$4. O(n^2)$$

$$5. O(n \ln(n))$$

$$6. O(\sqrt{n})$$

$$7. O(n!)$$

$$8. O(e^n)$$

$$9. O(\ln(\ln(n)))$$

Gesorteerd van groot naar klein:

7 - 8 - 3 - 2 - 4 - 5 - 1 - 6 - 9

#### 4. Algoritme:

input:  $n$

$t \leftarrow 0$

$s \leftarrow 0$

while  $s \leq n$  do

$t \leftarrow t + (1/s)$

$s \leftarrow s + 1$

$$C_1 = 1$$

$$C_2 = 1$$

$$C_3 =$$

$$C_4 = n+1$$

$$C_5 = 1$$

$$C_6 = 1$$

$$C = 1 + 1 + (n+1) \cdot (1+1)$$

$$= 2 + 2n + 2$$

$$= 2n + 4$$

$$f(n) = 2n + 4$$

$$f(n) \in O(n) \quad \text{want bij } c=3 \text{ en } n_0=5$$

$$\text{is } f(n) \leq 3n$$

#### 5 Algoritme:

input:  $n$

~~$t \leftarrow 1$~~

~~$s \leftarrow 1$~~

if  $n = 0$  do

return 1

else do

return  $n \cdot f(n-1)$

$$C_1 = 1$$

$$C_2 = 1$$

$$C_3 = n$$

$$C_4 = n$$

$$C = 1 + 1 + n$$

$$= 2n + 1$$



$$f(n) = 2n + 2$$

$f(n) \in O(n)$  want als we  $c=3$  nemen en  $n_0 = 3$  dan is  $f(n) \leq 3n$

6.

1. De while loop loopt precies  $n-1$  keer dus de functie voor dit algoritme is:

$$\begin{aligned} f(n) &= 3(n-1) + 2 \\ &= 3n - 3 + 2 \\ &= 3n - 1 \end{aligned}$$

$f(n) \in O(n)$  want als we  $c=3$  nemen en  $n_0 = 1$  dan is  $f(n) \leq 3n$

2. Algoritme:

input:  $n$

$s \leftarrow 0$

$i \leftarrow 0$

while  $i < n$  do

$j \leftarrow 0$

    while  $j < n$  do

$s \leftarrow s + 1$

$j \leftarrow j + 1$

$i \leftarrow i + 1$

$$C_1 = 1$$

$$C_2 = 1$$

$$C_3 = n - 1$$

$$C_4 = 1$$

$$C_5 = n - 1$$

$$C_6 = 1$$

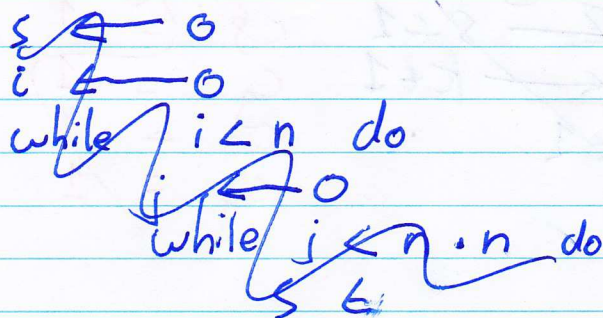
$$C_7 = 1$$

$$C_8 = 1$$

Als we de 1'tjes negeren zien we  
 $f(n) \approx (n-1)(n-1)$   
 $\approx n^2$

$P(n) \in O(n^2)$  omdat als we  $c = 1/2$  nemen en  $n_0 = 3$  dan  $P(n) \leq n^2$

3.



Dit is hetzelfde algoritme alleen de tweede while loop iterert  $n^2 - 1$  keer in plaats van  $n - 1$  keer.

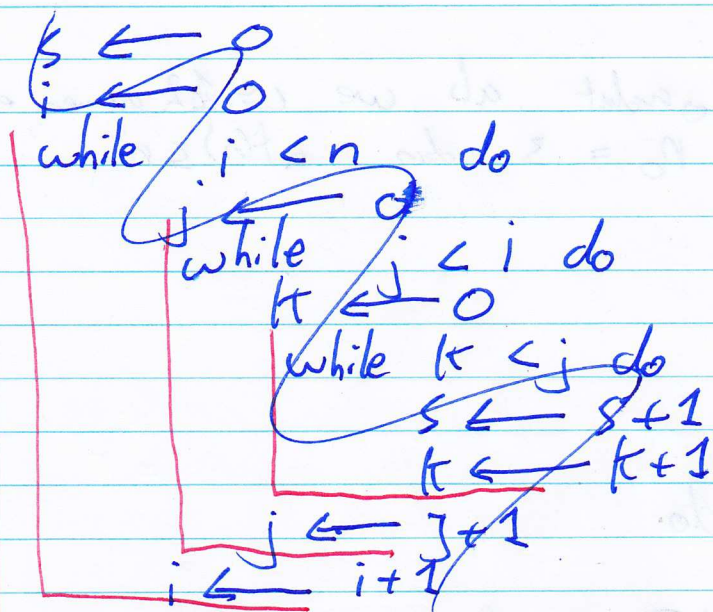
$$\begin{aligned} P(n) &= 3 + (n-1) \cdot (2n^2 - 1) \\ &= 3 + 2n^2 - 1 - 2n^2 + n \\ &= 2n^2 - n + 4 \end{aligned}$$

$P(n) \in O(n^3)$  omdat als we  $c = 1/2$  nemen en  $n_0 = 1$  dan  $P(n) \leq n^3$

4. Dit algoritme is met  $n > 1$  oneindig omdat de tweede while loop nooit stopt.



4.



$$\begin{aligned}
 C_1 &= 1 \\
 C_2 &= 1 \\
 C_3 &= n-1 \\
 C_4 &= 1 \\
 C_5 &= 1 \\
 C_6 &= 1 \\
 C_7 &= 1 \\
 C_8 &= 1 \\
 C_9 &= 1 \\
 C_{10} &= 1 \\
 C_{11} &= 1
 \end{aligned}$$

$$f(n) = 1 + 1 + ((n-1) \cdot (1 + (i-1) \cdot (1 + (j-1) \cdot 1 + 1) + 1) + 1) + 1$$

We kunnen de 1's weglaten voor het berekenen van de  $O(n)$ .

$$f(n) = n \cdot n!$$

5. De tweede while loop loopt  $1+2+3+4+5+\dots+n$  keer. Dit is  $\frac{n(n+1)}{2}$  keer. De derde while loop doet hetzelfde dus de totale functie loopt:

$$f(n) \approx \frac{n \left( \frac{n(n+1)}{2} + 1 \right)}{2} \quad (\text{keer})$$

$$= \frac{n \left( \frac{n^2 + n}{2} + 1 \right)}{2}$$

$$= \frac{n + \frac{n^3 + n^2}{2}}{2}$$

De <sup>groot</sup>

$f(n) \in O(n^3)$  want als we  $c = 2$  nemen  
en  $n_0 = 5$  dan is  $f(n) \leq 2n^3$