Afsnit 3.2

$f(x), g(x)$: funktioner.
Vi siger $f(x)$ er $O(g(x)$ hvis der findes konstanter k, C så

$$|f(x)| < C \cdot |g(x)|,$$

for alle (hele eller reelle) tal $x > k$.

Idé: $f(x)$ er et kompliceret udtryk, eller funktion der ikke kan beregnes præcist. $g(x)$ er et simpelt udtryk.
$g(x)$ vokser mindst lige så hurtigt som $f(x)$.

Korollar 1
Hvis $f_1(x)$ er $O(g(x))$ og $f_2(x)$ er $O(g(x))$
så er $f_1(x) + f_2(x)$ også $O(g(x))$.

Sætning 4
Lad $f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$, hvor $a_n \neq 0$
($f(x)$ er altså et vilkårligt polynomium af grad n)
Så er $f(x)$ $O(x^n)$ og x^n er $O(f(x))$.
Altså: $f(x)$ er $\Theta(x^n)$.

Eksempel.
$x^5 + 6x^4 + 3x^2 + 5$ er $O(x^5)$

$n^5 + 6n^4 + 3n^2 + 5$ er $O(n^5)$

$\log(x)^k$ er $O(x)$ for enhver konstant k.

Eksempel.
$x^2 \log(x)^3 + 3x^4 + 7x + 2\sqrt{x}$

er $O(x^4)$.

$f(x), g(x)$: funktioner.
Vi siger at $f(x)$ er $\Omega(g(x))$ hvis der findes positive konstanter k, C så

$$|f(x)| > C \cdot |g(x)|,$$

for alle (hele eller reelle) tal $x > k$.

($f(x)$ er $\Omega(g(x))$ hvis og kun hvis $g(x)$ er $O(f(x))$.)

Vi siger at $f(x)$ er $\Theta(g(x))$ hvis $f(x)$ er $O(g(x))$ og $f(x)$ er $\Omega(g(x))$.
($f(x)$ og $g(x)$ "vokser lige hurtigt")

Afsnit 3.3

Betragt en algoritme.
n : størrelsen af input.
$f(n)$: det største antal skridt algoritmen bruger hvis inputtet har størrelse n. (worst case)

Find et simpelt udtryk $g(n)$ så $f(n)$ er $O(g(n))$.

Vi siger at algoritmen har (tids-) kompleksitet $O(g(n))$.

procedure $bubblesort(a_1, \ldots, a_n$: reelle tal med $n \geq 2)$

for $i := 1$ **to** $n - 1$

    for $j := 1$ **to** $n - i$

        if $a_j > a_{j+1}$ **then** ombyt a_j og a_{j+1}

$\{ a_1, \ldots, a_n$ er nu i voksende rækkefølge$\}$