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 $f(x)$.

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

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

$f(x), g(x)$: 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")

procedure $linear\ search$(x:heltal, $a_1, \ldots, a_n$: forskellige heltal)
$i := 1$
while $i \leq n$ and $x \neq a_i$
. $i := i + 1$
if $i \leq n$ **then** $location := i$ **else** $location := 0$

{ hvis $location = 0$ så er x ikke i listen, ellers er $a_{location} = x$ }

procedure $binary\ search$(x: heltal, $a_1, \ldots, a_n$: voksende følge af heltal)
$i := 1$
$j := n$
while $i < j$
begin
. $m := \lfloor (i+j)/2 \rfloor$
. **if** $x > a_m$ **then** $i := m + 1$
. **else** $j := m$
end
if $x = a_i$ **then** $location := i$
else $location := 0$
$\{$ hvis $location = 0$ så er x ikke i listen, ellers er $a_{location} = x$ $\}$

procedure $change(c_1, \ldots, c_r, n$: positive hele tal)
{der skal udbetales n cents ved hjælp møntværdier $c_1 > c_2 > \ldots > c_n$, ($c_n = 1$)}
for $i := 1$ **to** r
. **while** $n \geq c_i$
. **begin**
. tilføj en mønt med værdi c_i til byttepengene
. $n := n - c_i$
. **end**

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 $O(g(n))$.