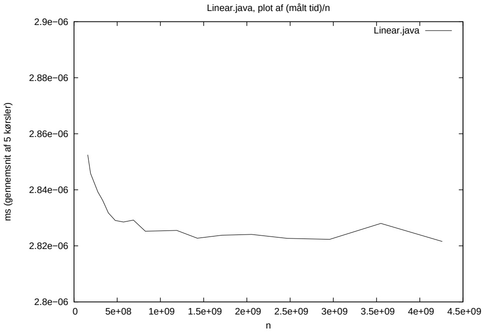
Asymptotisk analyse af algoritmers køretider

Analyse af køretid (RAM-modellen vs. virkeligheden)

```
public class Linear {
  public static void main(String[] args) {

    long time = System.currentTimeMillis();
    long n = Long.parseLong(args[0]);
    long total = 0;
    for(long i=1; i<=n; i++){
      total = total + 1;
    }
    System.out.println(total);
    System.out.println(System.currentTimeMillis() - time);
    }
}
```

$$T(n) = c_1 \cdot n + c_0$$



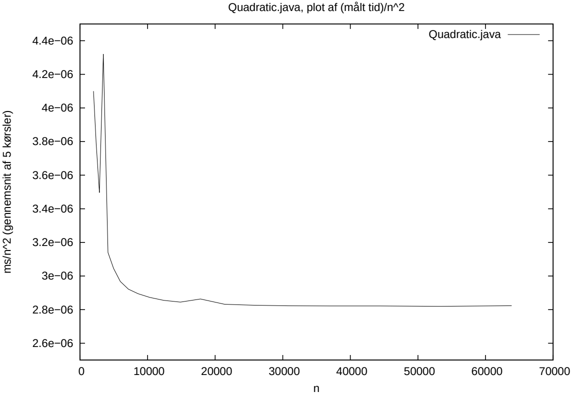
x-akse:
inputstørrelse n

y-akse:
(målt tid)$/n$

Analyse af tidsforbrug (RAM-modellen vs. virkeligheden)

```
for(long i=1; i<=n; i++){
  for(long j=1; j<=n; j++){
    total = total + 1;
  }
}
```

$$
\begin{aligned}
&T(n)\\
&= (c_2 \cdot n + c_1) \cdot n + c_0\\
&= c_2 \cdot n^2 + c_1 \cdot n + c_0
\end{aligned}
$$



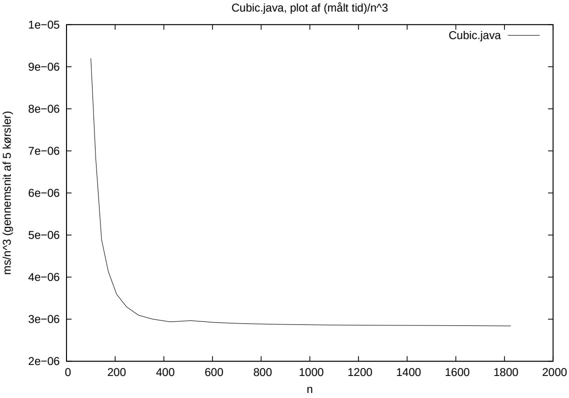
x-akse:
inputstørrelse n

y-akse:
(målt tid)$/n^2$

Analyse af tidsforbrug (RAM-modellen vs. virkeligheden)

```
for(long i=1; i<=n; i++){
  for(long j=1; j<=n; j++){
    for(long k=1; k<=n; k++){
      total = total + 1;
    }
  }
}
```

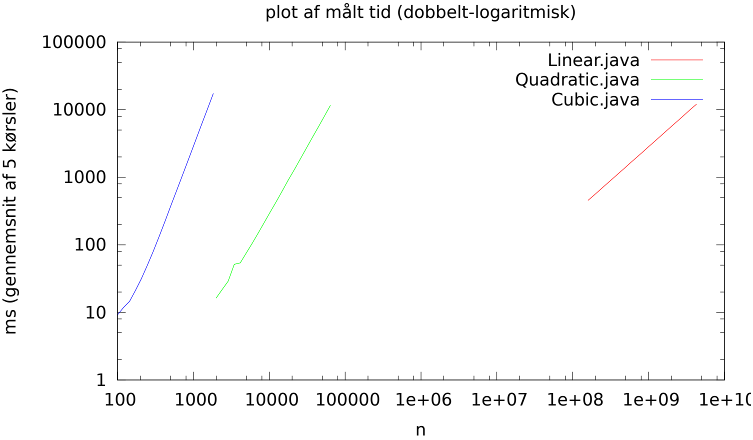
$$T(n)$$
$$= ((c_3 \cdot n + c_2) \cdot n + c_1) \cdot n + c_0$$
$$= c_3 \cdot n^3 + c_2 \cdot n^2 + c_1 \cdot n + c_0$$



x-akse:
inputstørrelse n

y-akse:
(målt tid)$/n^3$

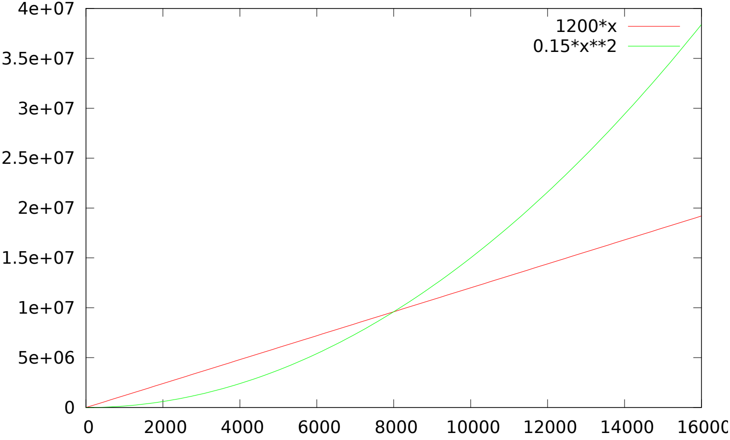
Linear vs. kvadratisk vs. kubisk



Multiplikative konstanter

Multiplikative konstanter ligegyldige hvis voksehastighed er forskellig:

$f(n) = 1200n$ $\qquad$ $g(n) = 0.15n^2$

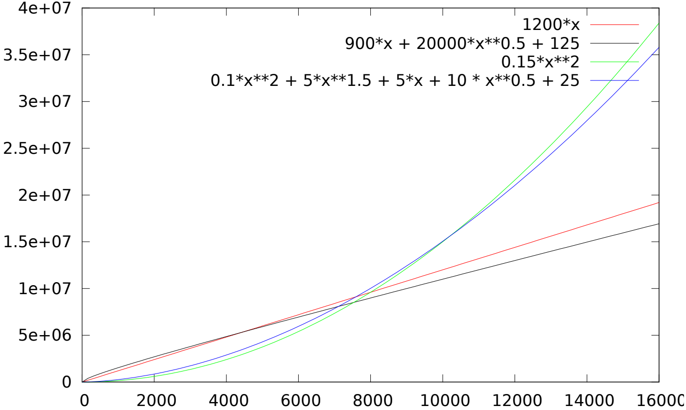


Dominerende led

Dominerende led bestemmer voksehastighed:

$$f(n) = 1200n \qquad g(n) = 0.15n^2$$

$$h(n) = 900n + 20000n^{0.5} + 125 \qquad k(n) = 0.1n^2 + 5n^{1.5} + 5n + 10n^{0.5} + 25$$



Asymptotisk notation

Vi ønsker at sammenligne funktioners essentielle voksehastighed på en måde så der ses bort fra multiplikative konstanter og ikke-dominerende led.

Vi ønsker for voksehastighed for funktioner analoger til alle fem klassiske ordens-relationer:

$$\leq \quad \geq \quad = \quad < \quad >$$

De vil, af historiske årsager, blive kaldt for:

$$O \quad \Omega \quad \Theta \quad o \quad \omega$$

Hvilket udtales således:

"Store O", "Omega", "Theta", "lille o", "lille omega"

Følgende definitioner har vist sig at fungere godt:

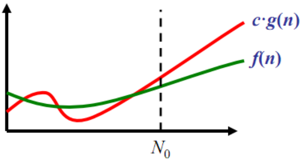
Store O

Definition: $f(n) = O(g(n))$

hvis $f(n)$ og $g(n)$ er funktioner $N \rightarrow R$ og

findes $c > 0$ og N_0 så *for alle* $n \geq N_0$:

$$f(n) \leq c \cdot g(n)$$



$\boxed{\text{Mening: } f \leq g \text{ i voksehastighed}}$

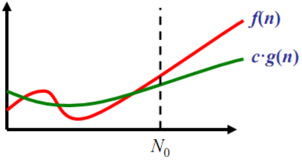
Store Omega

Definition: $f(n) = \Omega(g(n))$

hvis $f(n)$ og $g(n)$ er funktioner $N \to R$ og

findes $c > 0$ og N_0 så *for alle* $n \geq N_0$:

$$f(n) \geq c \cdot g(n)$$

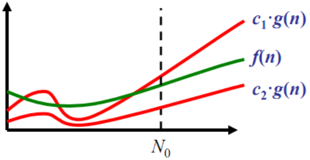


$\boxed{\text{Mening: } f \geq g \text{ i voksehastighed}}$

Theta

Definition: $f(n) = \theta(g(n))$

hvis $f(n)=O(g(n))$ og $f(n)=\Omega(g(n))$



$\boxed{\text{Mening: } f = g \text{ i voksehastighed}}$

Lille o

Definition: $f(n) = o(g(n))$

hvis $f(n)$ og $g(n)$ er funktioner $N \rightarrow R$ og

for alle $c > 0$, *findes* N_0 så *for alle* $n \geq N_0$:

$$f(n) \leq c \cdot g(n)$$

Mening: $f < g$ i voksehastighed

Lille omega

Definition: $f(n) = \omega(g(n))$

hvis $f(n)$ og $g(n)$ er funktioner $N \to R$ og

for alle $c > 0$, *findes* N_0 så *for alle* $n \geq N_0$:

$$f(n) \geq c \cdot g(n)$$

$\boxed{\text{Mening: } f > g \text{ i voksehastighed}}$

Asymptotisk analyse

De asymptotiske forhold mellem de fleste funktioner f og g kan afklares ved følgende sætninger:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = k > 0 \quad \Rightarrow \quad f(n) = \Theta(g(n))$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \quad \Rightarrow \quad f(n) = o(g(n))$$

Eksempler:

$$\lim_{n\to\infty} \frac{20n^2 + 17n + 312}{n^2} = \lim_{n\to\infty} \frac{20 + 17/n + 312/n^2}{1} = 20$$

$$\lim_{n\to\infty} \frac{20n^2 + 17n + 312}{n^3} = \lim_{n\to\infty} \frac{20/n + 17/n^2 + 312/n^3}{1} = 0$$

Asymptotisk analyse

Derudover er det godt at vide følgende:

$$\lim_{n\to\infty} \frac{n^a}{b^n} = 0 \text{ for alle } a > 0 \text{ og } b > 1$$

(Dette kan f.eks. vises ved at bruge l'Hôspitals regel fra MM529/MM501 i alt $\lceil a \rceil$ gange). For $c > 1$ og $d > 0$, sæt $n = \log_c(N)$ og $b = c^d$. Så fås følgende variant:

$$\lim_{N\to\infty} \frac{(\log_c N)^a}{N^d} = 0 \text{ for alle } a, d > 0 \text{ og } c > 1$$

Dvs. enhvert polynomium er $o()$ af enhver exponentialfunktion, og enhver logaritme (selv opløftet i enhver potens) er $o()$ af enhvert polynomium.

Eksempelvis giver dette at:

$$\lim_{n\to\infty} \frac{n^{100}}{2^n} = 0 \Rightarrow n^{100} = o(2^n) \quad \text{og} \quad \lim_{n\to\infty} \frac{(\log n)^3}{n^{0.5}} = 0 \Rightarrow (\log n)^3 = o(n^{0.5})$$