$T(n)$ definition alle $\bar{I}$ mit $(\bar{I}) = n$

(1) Best case: $T_{bc}(n) = \min_{|I| = n} T(I)$

$t_j = 1$ für alle j

$\downarrow$

Laufzeit für bestcase für Problemgröße n

$\sum_{j=2}^{n} t_j = n - 1$

t_j... Anz der Tests (5) für j

$T_{bc}(n) = a \cdot n + b$

(2) Worst Case: Wir schauen das Maximum an

$T_{wc}(n) = \max_{|I| = n} T(I)$

$t_j = j$

$$\sum_{j=2}^{n} t_j = \sum_{j=2}^{n} j = \left(\sum_{j=1}^{n} j\right) - 1$$

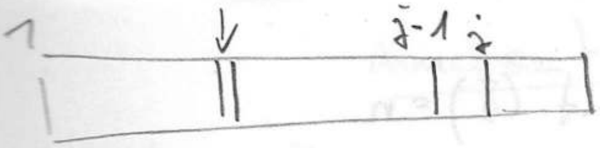
$$\rightarrow \frac{n \cdot (n+1)}{2} - 1$$

$T_{wc}(n) = an^2 + bn + c$

$$\sum_{j=2}^{n} (t_j - 1) = \frac{n(n-1)}{2}$$

(3) Average Case: Summe aller Laufzeiten von Größe n div. durch ?

$$T_{ac}(n) = \sum_{|I|=n} T(I) \cdot \frac{1}{|\{I\}| \cdot |I| = n}$$



$t_j = \frac{j}{2}$ $\quad \sum_{j=2}^{n} j/2 = \left(\frac{n\cdot(n+1)}{2} - 1\right)/2$! Anschauen

$T_{ac}(n) = an^2 + bn + c$

Avarage ~~Ca~~ Case: meistens schwer zu lösen
also die Analyse
Verteilungen müssen
berücksichtigt werden

$$T_{bc}(n) \le T_{avg}(n) \le T_{wc}(n)$$

$$\exists I_{|I|=n},\ T(I) \le T_{avg}(n)$$

$$\exists I_{|I|=n}\ T(I) \ge T_{avg}(n)$$

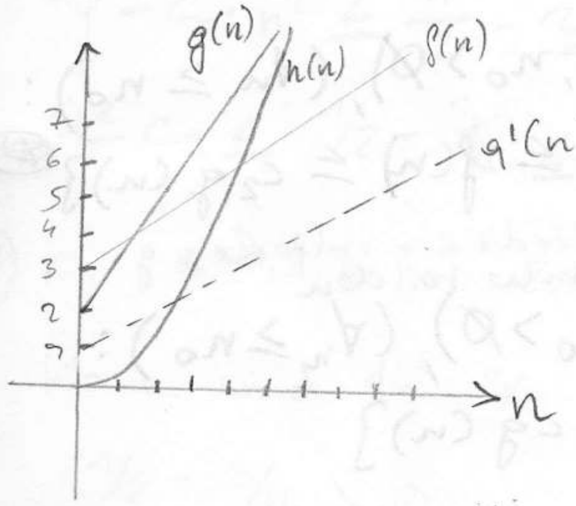
Asymptotische Analyse

Wachstumsverhalten!

Laufzeitfunktionen

Progr. 1 : $f(n) = n+3$
— " — 2 : $g(n) = 2n+2$ } lang
— " — 3 : $h(n) = n^2$ } Graphik

$h(n)$ ... ist am schnellsten



Dinge wie zB Taktfrequ. sollen eigentl. keine Rolle bei den spielen, es soll skalierbar sein...

- Wir sind für Laufzeiten für große n interessiert
  → Asymptotisches Wachstum, für $n \to \infty$
- Eine Skalierung um einen konst. Faktor ändert nichts am asympt. Wachstum!
- „$f(n)$ wächst ungefähr gleich wie $g(n)$" wenn

$$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$$

für konstanten $c_1, c_2 > 0$

Notationen WICHTIG!

Groß Oh	Groß Omega	Groß Theta
O für obere Schranke	Ω für untere Schranke	Θ genaues Wachstum

Sei $g: \mathbb{N}_0 \to \mathbb{N}_0$ eine Funktion.
Wir definieren die folg. Mengen der Funktionen.

$$\Theta(g(n)) = \{ f(n) \mid (\exists c_1, c_2, n_0 > 0), (\forall n \geq n_0): 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)\}$$

Menge aller Funktionen für die ein Vielfaches g eine obere Schranke bilden

$$O(g(n)) = \{ f(n) \mid (\exists c, n_0 > 0), (\forall n \geq n_0): 0 \leq f(n) \leq c g(n)\}$$

$$\Omega(g(n)) = \{ f(n) \mid (\exists c, n_0 > 0), (\forall n \geq n_0): 0 \leq c g(n) \leq f(n)\}$$

$$\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$$

$$f(n) \in \Theta(g(n)) \rightsquigarrow f(n) = \Theta(g(n))$$

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

$\forall h \in O(g(n))$ gilt $h(n) \in O(f(n))$

Bsp: $f(n) = \frac{n^2}{2} - 3n \in \Theta(n^2) \quad (g(n) = n^2)$

① Auf der Suche nach c_1, c_2, n_0

$$0 \leq c_1 \cdot n^2 \leq \frac{n^2}{2} - 3n \leq c_2 \cdot n^2, \quad \forall n \geq n_0 \;\Big|\; : n^2$$

$$0 \leq c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2$$

1) c_1: $n = 1, 2, 3$ $\quad -2{,}5,\ -1,\ -\frac{1}{2} \ldots$

Wann wird $\frac{1}{2} - \frac{3}{n} > 0$

$\frac{1}{2} - \frac{3}{n} > 0$:

$\Rightarrow n = 6 \quad \Rightarrow \quad n \geq 7$

$c_1 = \frac{1}{2} - \frac{3}{7} = \frac{1}{14} \Rightarrow n_0 = 7$

2) c_2: $n = 1, 2, 3 \ldots$ weil $\frac{1}{2} - \frac{3}{n} \leq \frac{1}{2} \quad \forall n \geq 1$

$\Rightarrow c_2 = \frac{1}{2}, \quad n_0 = 1$

$n_0 = 7, \quad c_1 = \frac{1}{14}, \quad c_2 = \frac{1}{2}$

Bsp: $f(n) = a \cdot n + b \in \Theta(n)$

Kubische steigen mehr an als quadr.

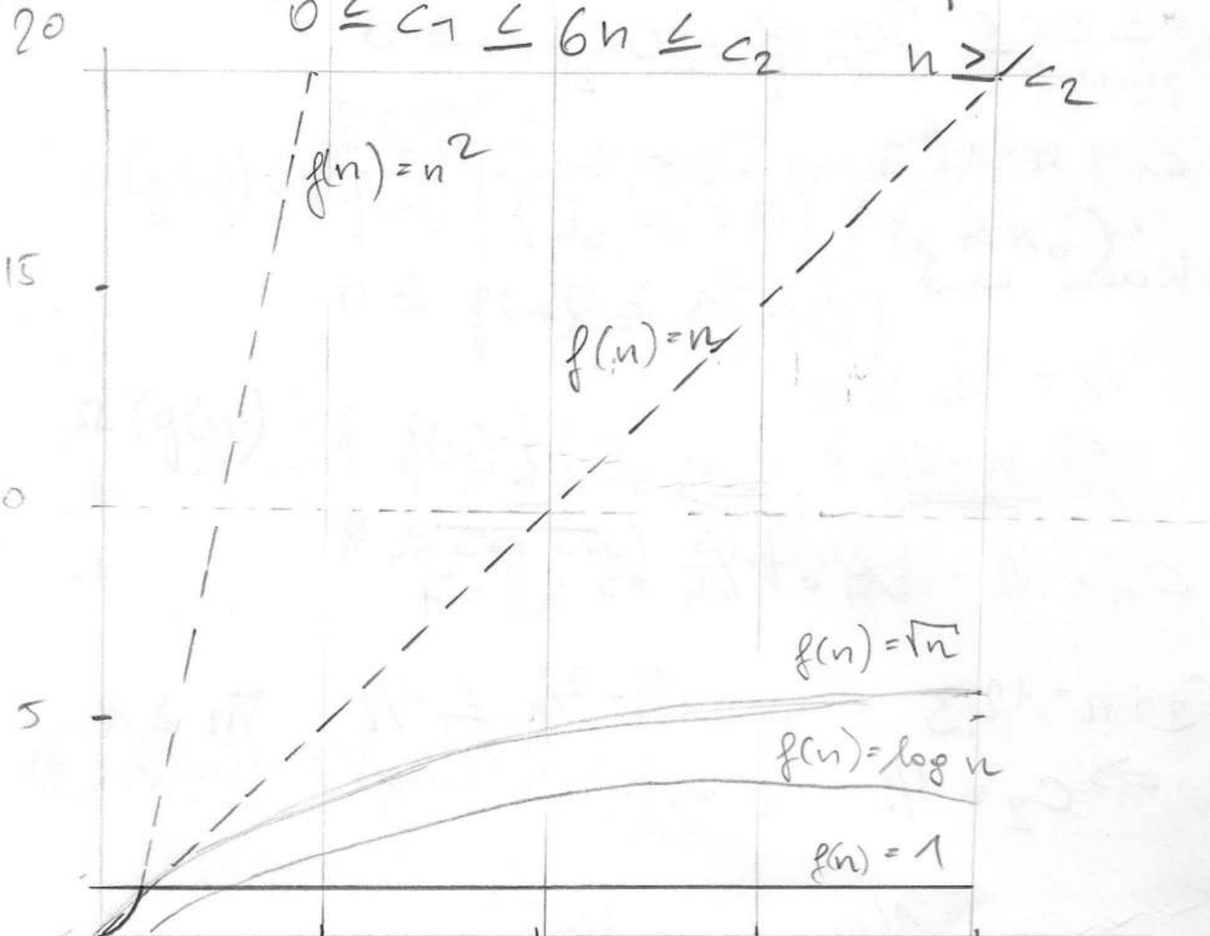
Bsp: $f(n) = 6 \cdot n^3 \overset{?}{\underset{\notin}{=}} \Theta(n^2)$

Ang. $6 \cdot n^3 \in \Theta(n^2)$

$\Rightarrow \exists c_1, c_2, n_0 > 0$

$0 \leq c_1 \cdot n^2 \leq 6n^3 \leq c_2 \cdot n^2 \mid 1/n^2$

$0 \leq c_1 \leq 6n \leq c_2 \qquad n \geq c_2$



$\Theta(1)$ .... konst. Laufzeit

$\Theta(n)$ ... lineare Laufzeit

$\Theta(\sqrt{n})$

$\Theta(\log n)$ ... $\log_2 n = \mathrm{ld}\, n$

Testvorbereitung! 13. Okt 2009

Fortsetzung asymptotische Analyse ...

Welche Aussagen ~~treffen~~ zu?

- $g(n) = O(f(n))$ und $h(n) = O(g(n))$ impliziert $h(n) = O(f(n))$

obere Schranke von

obere Schr. von

$g(n) \overset{n_0^1}{\le} c_1 \cdot f(n)$, $h(n) \le c_2 \cdot g(n)^{n_0^2}$

$\Rightarrow h(n) \le c_1 \cdot c_2 \cdot f(n)$

es gilt ab dem $\max(n_0^1, n_0^2)$

- $g(n) = O(f(n))$ genau dann wenn $f(n) = \Omega(g(n))$

ob. Schr.

untere Schr

$g(n)$ liegt immer unter $f(n)$

$g(n) \le c_1 \cdot f(n) \Rightarrow f(n) \ge \frac{1}{c_1} g(n)$

aus li folgt re und umgekehrt

- $g(n) = \Theta(f(n))$ impliziert
$f(n) = \Omega(g(n))$ oder (nicht $f(n) = O(g(n))$)

wahr wenn ω oder w

$g(n) = \Theta(f(n))$ wachsen ca. gleich an

$f(n) = \Omega(g(n))$ $g(n)$ ist untere Schr. von $f(n)$

weil diese zwei wahr sind, brauchen wir den Teil nach dem ODER nicht

also gilt die Aussage

$g(n) = \Theta(f(n)) \Rightarrow f(n) = \Theta(g(n))$

- $(h(n) = O(f(n) \cdot g(n))$ und $h(n) = O(g(n)))$ impliziert $(f(n) = \Theta(n))$

$h(n)$ obere Schranke $f(n) \cdot g(n)$

$g(n)$ ist ob. Schranke von $h(n)$

wenn g alleine schon Schranke von h kann ich es mit einem beliebig $f(n)$ multiplizieren.

diese Aussage gilt nicht

$h(n) = g(n) = 1$, $\underline{f(n)} = n^2 \notin O(n)$

$f(n) \cdot g(n) = n^2$ dadurch ist $h(n)$ nach oben hin beschränkt

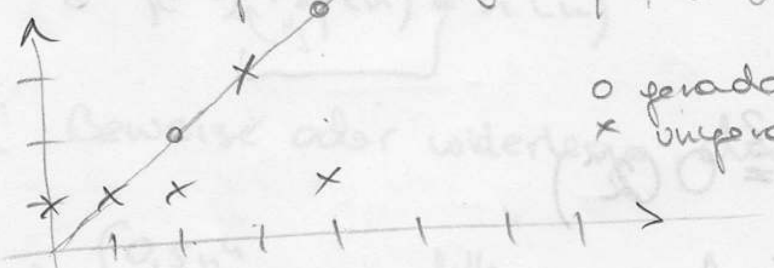
• $f(n) = O(g(n))$ oder $g(n) = \Omega(f(n))$

bedeutet: immer obere Schranke — immer obere Schranke

kann nicht sein weil man hier beliebige Kurven zeichnen könnte.

"auseinanderdriftende Funktionen"

$$f(n) = \begin{cases} 1, & n \text{ gerade} \\ n, & n \text{ ungerade} \end{cases} \times \quad g(n) \begin{cases} n, & n \text{ gerade} \\ 1, & n \text{ ungerade} \end{cases}$$



Bsp: $f(n) = 2n^2 \cdot \log(3n) = O(n^{5/2})$

$\exists c, n_0 > 0 \ \forall n \geq n_0: 0 \leq f(n) \leq c \cdot n^{5/2}$

Regel: $f(n) = f_1(n) \cdot f_2(n)$ und man weiß dass f_1 nach oben hin durch FkT g_1 beschränkt ist und f_2 durch g_2

$\wedge f_1(n) = O(g_1(n)) \wedge f_2(n) = O(g_2(n))$

$\Rightarrow f(n) = O(g(n)),\ g(n) = g_1(n) \cdot g_2(n)$

wir können unsere FkT jetzt zerlegen ...

$f(n) = f_1(n) \cdot f_2(n)$; $f_1(n) = 2n^2$

$f_2(n) = \log(3n)$ — gleiches Größenwachstum

$c = 2$ ✓

Es reicht: $f_1(n) = O(n^2)$

nach oben hin beschränkt ist durch $f_2(n) = O(n^{1/2})$

$\log 3n \leq c \cdot \sqrt{n}$, $\forall n \geq n_0$

Es reicht: $f_1(n) = O(n^2)$ $\quad f_1(n) = n^2$

$f_2(n) = O(n^{1/2})$ $\quad f_2(n) = \log 3$

$c = 2$

$\underline{\log 3n} \leq c \cdot \sqrt{n}$ $\qquad \log 3 + \log n \leq c \cdot \sqrt{n}$

$\underline{\log n} \leq \underline{c \cdot \sqrt{n}} - \underbrace{\log 3}_{>1,\,<2}$

$n =$	1	2	4	8	16	32	64	128
$\sqrt{n} =$	1	1,4…	2	2,…	4	5,…	8	11
$\log n =$	$\emptyset$	1	2	3	4	5	6	7

2^4

<u>Bsp:</u> $\quad 3^n \stackrel{?}{=} O(2^n)$

Ang. die Aussage gilt:

$\exists c, n_0 \quad \forall n \geq n_0$

$3^n \leq c \cdot 2^n \quad | \log$

$\log 3^n \leq \log(c \cdot 2^n)$

$n \cdot \underbrace{\log 3}_{>1 \wedge <2} \leq \log c + n \cdot \cancel{\log 2}^{\,2}$ $\quad$ | n auf die andere Seite bringen

$n \cdot \underbrace{(\log 3 - 1)}_{\substack{>1 \\ >0}} \leq \underbrace{\log c}_{const.}$

$n \cdot c_1 \leq \log c$

GILT NICHT DIE AUSSAGE!

Interpretation von Gleichungen

zB: $f(n) = g(n) + \Theta(n)$

Bedeutet: $\exists h(n) \in \Theta(n)$, sodass $f(n) = g(n) + h(n)$

zB: $f(n) + O(n) = \Theta(n^2)$

Für alle $g(n) \in O(n)$ gibt es ein $h(n) \in \Theta(n^2)$
Sodass $f(n) + g(n) = h(n)$

Beispiel: Beweise oder widerlege, dass $f(n) = \Theta(n^4)$.

$$f(n) = \begin{cases} 0{,}5n^4 & \text{falls } n \text{ prim} \\ n^4 - n \log n & \text{falls } n \neg\text{prim} \end{cases}$$

Beachte, dass zu einem vollst. Beweis gegebenf. auch geeignete Werte für die verwendete Konst. anzugeben sind.

$f_1(n) = \frac{1}{2} \cdot n^4$ // würd stimmen (gl. Wachstum)

$f_2(n) = n^4 - \underbrace{n \cdot \log n}_{\underbrace{\log n \leq n}_{\leq n^2}}$ $n^4 - n^2$

$f_1(n) = \Theta(n^4)$ // immer von unten durch n^4 beschränkt

$f_2(n) = \Theta(n^4)$

$f(n) = \Theta(n^4)$

für $f_1 \Rightarrow c_1, c_2$: $c_1 = 1/2$ (da n^4 untere Schranke)

$c_2 = 1$

$n_0 = 1$

für f_2: $\underbrace{c_1 \cdot n^4 \le n^4 - n \log n \le c_2 \cdot n^4}_{/n^4}$ $\quad c_2 = 1$

$$c_1 \le 1 - \frac{\log n}{n^3}$$

$$c_1 \le 1 - \frac{n}{n^3}$$

$$c_1 \le 1 - 1/n^2$$

wenn $n \ge 2$: $c_1 = 3/4$ ist Ungl. erfüllt

Ergebnis: $c_1 = 3/4$ $\quad c_2 = 1$ $\quad n_0 = 2$

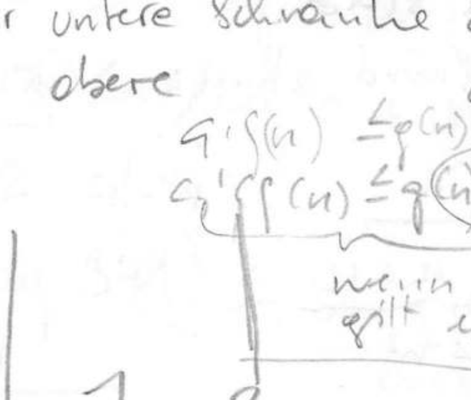
für untere Schranke $c_1 = 1/2$

obere $c_2 = 3/4$

$c_1 \cdot f(n) \le g(n)$

$c_1' \cdot f(n) \le g(n)$ $\quad n_0 = 2$

wenn ich das Max. nehme gilt es für beide



$g(n) \le c_1 \cdot h(n)$

$g(n) \le c_2 \cdot h(n)$

$$g(n) \le \max(c_1, c_2) \cdot h(n)$$

Welche Laufzeiten in Θ-Notation haben die folgenden Programmcodefragmente in Abhängigkeit von n?

(i)
```
p = n + 51;
wiederhole
    k = n³ − p;
    p = p − 1;
bis p < 4
```

Größenwachstum = n

$\Theta(n)$

(ii)
```
p = n;
solange p > 0
    für i = 1, ..., n {
        k = k + i;
    }
    p = [p/2];   // halbiert und rundet ab
}
```

n Schritte

Wenn man fortl. halbiert kommt man irgendwann auf Ø deshalb → Logarithmus

$n = 2^m$ dann wird die Schleife $\log n + 1$ durchlaufen. (wenn es eine der Potenz ist)
2 4 8 16 32 ...

$n = 2^m$ $\quad \exists m : 2^m \leq n \leq 2^{m+1}$

$\underbrace{\log n}_{m+1} + 1$

$\Theta\ n \cdot \log n$