

Das große “O” (1)

- ▶ **Aufwandsklasse** $O(g)$ für $g: \mathbb{N} \rightarrow \mathbb{R}^+$ enthält alle Funktionen $f: \mathbb{N} \rightarrow \mathbb{R}^+$ mit

$$f(n) \leq c \cdot g(n) \text{ für alle } n \geq n_0$$

c, n_0 : konstant

- ▶ $O(g)$ beschreibt alle Probleme, die eine algorithmische Lösung op besitzen mit $T^{op} \in O(g)$
- ▶ Obere Schranke, bei der konstante Faktoren und kleine Eingaben nicht zählen.

Das große “O” (2)

► **Idee:** wähle g einfach und einprägsam, z.B.

$$\log n, n, n \cdot \log n, n^2, n^3, \dots 2^n, \dots$$

- $\log n$: **logarithmisch**
- n : **linear**
- n^2 : **quadratisch**
- n^3 : **kubisch**
- 2^n : **exponentiell**

Beispiele

- ▶ $T^{\text{mergesort}} \in O(n \cdot \lg n)$
- ▶ $T^{\text{quicksort}} \in O(n^2)$
- ▶ $T^{\text{xmpl}}(n) \leq 1/3n \cdot \lg n + 22n + 1/2n^2 + 7$
 $\rightsquigarrow T^{\text{xmpl}} \in O(n^2),$

d.h. *xmpl* hat quadratischen Aufwand.

Zusammenhang zwischen verschiedenen Aufwandsklassen

Beobachtung

$$O(\log n) \subset O(n) \subset O(n \log n) \subset O(n^2) \subset O(n^3) \subset \dots \subset O(2^n)$$

(vergl. Übung)

Linear, quadratisch und exponentiell im Vergleich

n	n^2	2^n
10	100	1024 $\sim 10^3$
20	400	1048576 $\sim 10^6$
30	900	1073741824 $\sim 10^9$
40	1600	10999511627776 $\sim 10^{13}$
50	2500	1125899906842624 $\sim 10^{15}$
60	3600	1152921504600686976 $\sim 10^{18}$
70	4900	1180591620717411303424 $\sim 10^{21}$

ohne Gewähr

-
- ▶ Das Wachstum dieser Funktionen ist bekannt, aber was bedeutet das schon bei permanent wachsender Rechnerleistung?

Mit Supercomputer gegen Aufwand (?)

- ▶ **Szenario:** entwickle ein Programm, das bei Eingabegröße n gerade n | n^2 | 2^n FLOPs benötigt und miete für 3 Stunden ($\sim 10^4$ Sek.) einen Supercomputer mit 100 Tera-FLOPs pro Sekunde (10^{14} FLOPs/Sek.).
- ▶ Welche Eingabegröße lässt sich bearbeiten?

n	n^2	2^n
10^{18}	10^9	60

Mit Supercomputer gegen Aufwand (?)

- ▶ **Szenario:** entwickle ein Programm, das bei Eingabegröße n gerade n | n^2 | 2^n FLOPs benötigt und miete für 3 Stunden ($\sim 10^4$ Sek.) einen Supercomputer mit 100 Tera-FLOPs pro Sekunde (10^{14} FLOPs/Sek.).
- ▶ Welche Eingabegröße lässt sich bearbeiten?

n	n^2	2^n
10^{18}	10^9	60
10^{21}	$32 \cdot 10^9$	70 (*)

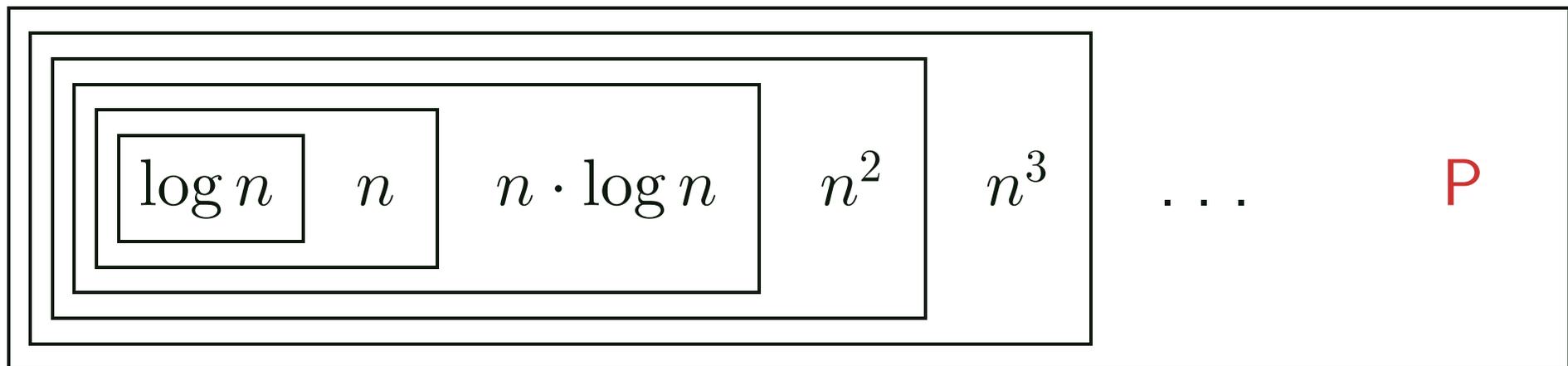
(*): bei Rechenzeitverbesserung um Faktor 1000

Anmerkungen

- ▶ Statt schlechtestem Fall macht auch Durchschnitt viel Sinn (ist aber meist viel schwerer zu ermitteln).
- ▶ Standardmaß für Zeitaufwand ist die Zahl der Rechenschritte einer Turing-Maschine.
- ▶ Zur Ermittlung wird aber meist eine Aktivität gezählt, die konstant beschränkt viele Rechenschritte braucht.
- ▶ Gleichungsanwendung ist nicht immer durch konstant beschränkt viele Rechenschritte realisierbar.
- ▶ CE-S-Berechnungen sind i.a. nichtdeterministisch.

Polynomieller Aufwand: die Klasse P

- ▶ Probleme in P können mit herkömmlichen Rechnern in verfügbarer Zeit gelöst werden (wenn der Exponent nicht zu groß ist)



Beispiele

Problem aus P	Aufwand
Suchen in balancierten Bäumen	$\log n$
Suchen in Wörtern, Transponieren, Zählen, Filtern	n
Sortieren durch Mischen	$n \cdot \log n$
Sortieren durch Einsortieren, <i>quicksort</i>	n^2
Matrizenmultiplikation, Wortproblem kontextfreier Sprachen	n^3

Die Klassen P und NP

P enthält die **Probleme**, die mit herkömmlichen Computern zeitgerecht lösbar sind

NP enthält eine Vielzahl praktisch relevanter Probleme (wie Tourenplanung, Maschinenbelegung, Stundenplanung, Lagerhaltung, . . .)

Definition von NP und P (1)

Entscheidungsproblem

Abbildung der Form $dp: A^* \rightarrow BOOL$

Lösung von dp

CE-S-Operation $sol: A^* \rightarrow BOOL$, so dass für alle $w \in A^*$ gilt:

$$dp(w) = T \text{ g.d.w. } sol(w) \overset{*}{\longleftrightarrow} T$$

(aber: $dp(w) = F$ g.d.w. $sol(w) \overset{*}{\longleftrightarrow} T$ nicht gilt)

Definition von NP und P (2)

$dp \in NP$, falls eine Lösung sol und ein $k \in \mathbb{N}$ existieren mit $T^{sol} \in O(n^k)$

$dp \in P$, falls zusätzlich für alle $w \in A^*$ gilt:

$$dp(w) = T \text{ und } sol(w) \overset{*}{\longleftrightarrow} t \text{ impl. } t = T$$

offensichtlich: $P \subseteq NP$

offen: $NP \subseteq P$