Beispiel 2

$r := 2$
for $i := 1$ **to** n **do** $r := r^2$ **od**
co `das Ergebnis ist` 2^{2^n} **oc**

- Zeitbedarf:
  - uniform: n Schritte
  - logarithmisch: $1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1 = \Theta(2^n)$
- Platzbedarf:
  - uniform: $\mathcal{O}(1)$
  - logarithmisch: 2^n

6. Wachstumsverhalten von Funktionen

f, g seien Funktionen von $\mathbb{N}_0$ nach $\mathbb{R}_+$.

- $g = \mathcal{O}(f)$ [auch: $g(n) = \mathcal{O}(f(n))$ oder $g \in \mathcal{O}(f)$] gdw.

$$(\exists c > 0 \exists n_0 \in \mathbb{N}_0 \forall n \geq n_0)[g(n) \leq c \cdot f(n)]$$

- $g = \Omega(f)$ [auch: $g(n) = \Omega(f(n)$ oder $g \in \Omega(f)$] gdw.

$$(\exists c > 0 \exists n_0 \in \mathbb{N}_0 \forall n \geq n_0)[g(n) \geq c \cdot f(n)]$$

- $g = \Theta(f)$ gdw. $g = \mathcal{O}(f)$ und $g = \Omega(f)$

f, g seien Funktionen von $\mathbb{N}_0$ nach $\mathbb{R}_+$.

- $g = o(f)$ gdw.

$$(\forall c > 0 \exists n_0 \in \mathbb{N}_0 \forall n \geq n_0)[g(n) \leq c \cdot f(n)]$$

- $g = \omega(f)$ gdw.

$$(\forall c > 0 \exists n_0 \in \mathbb{N}_0 \forall n \geq n_0)[g(n) \geq c \cdot f(n)]$$

- $g = \Omega_\infty(f)$ gdw.

$$(\exists c > 0)[g(n) \geq c \cdot f(n) \text{ für unendlich viele } n]$$

- $g = \omega_\infty(f)$ gdw.

$$(\forall c > 0)[g(n) \geq c \cdot f(n) \text{ für unendlich viele } n]$$

Beispiel 3

- n^3 ist nicht $\mathcal{O}\left(\frac{n^3}{\log n}\right)$.
- $n^3 + n^2$ ist nicht $\omega(n^3)$.
- $100n^3$ ist nicht $\omega(n^3)$.

Bemerkung:
Die Groß-O-Notation wurde von D. E. Knuth in der Algorithmenanalyse eingeführt, siehe z.B.

Donald E. Knuth:
Big omicron and big omega and big theta.
SIGACT News **8**(2), pp. 18–24, ACM SIGACT, 1976

Sie wurde ursprünglich von Paul Bachmann (1837–1920) entwickelt und von Edmund Landau (1877–1938) in seinen Arbeiten verbreitet.

	Problemgröße n					
Wachs-tums-rate	10	20	30	40	50	60
n	.00001 Sekunden	.00002 Sekunden	.00003 Sekunden	.00004 Sekunden	.00005 Sekunden	.00006 Sekunden
n^2	.0001 Sekunden	.0004 Sekunden	.0009 Sekunden	.0016 Sekunden	.0025 Sekunden	.0036 Sekunden
n^3	.001 Sekunden	.008 Sekunden	.027 Sekunden	.064 Sekunden	.125 Sekunden	.216 Sekunden
n^5	.1 Sekunden	3.2 Sekunden	24.3 Sekunden	1.7 Minuten	5.2 Minuten	13.0 Minuten
2^n	.001 Sekunden	1.0 Sekunden	17.9 Minuten	12.7 Tage	35.7 Jahre	366 Jahrhdte
5^n	.059 Sekunden	58 Minuten	6.5 Jahre	3855 Jahrhdte	2×10^8 Jahrhdte	1.3×10^{13} Jahrhdte

Beispiel 4 (Mergesort)

$$
\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + cn \\
&= cn + 2T\left(\frac{n}{2}\right) \\
&= cn + 2\left(c\frac{n}{2} + 2T\left(\frac{n}{4}\right)\right) \\
&= cn + cn + 4T\left(\frac{n}{4}\right) \\
&\approx cn \log_2 n \text{ (nur genau für Zweierpotenzen)}
\end{aligned}
$$

Methoden zur Lösung von Rekursionsgleichungen

1. Multiplikatorenmethode
2. Lineare homogene Rekursionsgleichungen können mit Hilfe des charakteristischen Polynoms gelöst werden
3. Umwandlung inhomogener Rekursionsgleichungen in homogene
4. Erzeugendenfunktionen
5. Transformation des Definitions- bzw. Wertebereichs
6. ...

Es gibt keinen vollständigen Satz von Methoden.

7.1 Multiplikatoren

Sei $f_1 = 1$, $f_n = 2f_{n-1} + n$ für $n \geq 2$.

$$\left.\begin{array}{rll} f_n = 2f_{n-1} + n & | \cdot 1 & \\ f_{n-1} = 2f_{n-2} + n - 1 & | \cdot 2 & \\ \vdots & \vdots & \\ f_2 = 2f_1 + 2 & | \cdot 2^{n-2} & \\ f_1 = 1 & | \cdot 2^{n-1} & \end{array}\right\} \begin{array}{l} f_n = 2^{n-1} f_1 + \sum_{i=0}^{n-2} 2^i (n-i) \\ \quad = 2^{n+1} - n - 2 \end{array}$$

Durch Addieren aller Gleichungen erhalten wir:

$$f_n = 2^{n-1} + \sum_{i=0}^{n-2} 2^i(n-i) = \underbrace{2^{n-1}}_{(1)} + \underbrace{n\sum_{i=0}^{n-2} 2^i}_{(2)} - \underbrace{\sum_{i=0}^{n-2} i \cdot 2^i}_{(3)}$$

Term (2) (geometrische Reihe):

$$n\sum_{i=0}^{n-2} 2^i = n(2^{n-1} - 1)$$

Term (3) (mit der Substitution $n-2=k$ und x für 2):

$$\begin{aligned}\sum_{i=0}^{k} ix^i &= \sum_{i=1}^{k} ix^i = x\sum_{i=1}^{k} ix^{i-1} \\ &= x\sum_{i=1}^{k} \frac{dx^i}{dx} = x\frac{d}{dx}\sum_{i=1}^{k} x^i \\ &= x\frac{d}{dx}\left(\frac{x^{k+1}-1}{x-1}\right) \\ &= \frac{kx^{k+2}-x^{k+1}(k+1)+x}{(x-1)^2}\end{aligned}$$

Einsetzen von $k=n-2$ und $x=2$ ergibt:

$$\sum_{i=0}^{n-2} i2^i = 2^n(n-2)-2^{n-1}(n-1)+2 = 2^n n-2^{n+1}-2^{n-1}n+2^{n-1}+2$$

$(2) - (3)$:

$$n\sum_{i=0}^{n-2} 2^i - \sum_{i=0}^{n-2} i2^i = n \cdot (2^{n-1} - 1) - 2^n(n-2) + 2^{n-1}(n-1) - 2$$
$$= 2^{n-1}(2n-1) - 2^n(n-2) - n - 2$$

und schließlich $(1) + (2) - (3)$:

$$2^{n-1} + n\sum_{i=0}^{n-2} 2^i - \sum_{i=0}^{n-2} i2^i = 2^{n-1}(1 + 2n - 1) - 2^n(n-2) - n - 2$$
$$= 2^{n+1} - n - 2$$

7.2 Charakteristisches Polynom

Sei

$$f_0 = 0$$
$$f_1 = 1$$
$$f_n = f_{n-1} + f_{n-2} \text{ für } n \geq 2 \, .$$

Es handelt sich hier um eine lineare homogene Rekursionsgleichung zweiter Ordnung.

Ansatz: $f_n := a^n$ für ein unbekanntes a.
Dann muss gelten: $a^n - a^{n-1} - a^{n-2} = 0$. Da hier $a \neq 0$:

$$a^2 - a - 1 = 0; \text{ also } a_{1/2} = \frac{1 \pm \sqrt{5}}{2}$$

Falls $f_n = a_1^n$ und $f_n = a_2^n$ Lösungen der Rekursionsgleichung sind, dann auch $f_n = c_1 a_1^n + c_2 a_2^n$, für beliebige Konstanten c_1 und c_2.

$f_1 = 1$ und $f_0 = 0$ liefern zwei Gleichungen für c_1 und c_2, mit der Lösung:

$$f_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right]$$

Satz 5

Sei $p(x)$ das charakteristische Polynom zur (linearen homogenen) Rekursionsgleichung

$$p_0 f_n + p_1 f_{n-1} + \cdots + p_k f_{n-k} = 0 \qquad (1)$$

mit den konstanten Koeffizienten p_i. Seien r_i, $i = 1, \ldots, m$ die (i.a. komplexen) Wurzeln von $p(x)$, jeweils mit Vielfachheit m_i. Dann ist die allgemeine Lösung der Rekursionsgleichung (1) gegeben durch

$$f_n = \sum_{i=1}^{m} \left(r_i^n \sum_{j=0}^{m_i - 1} c_{ij} n^j \right),$$

mit Konstanten c_{ij}.