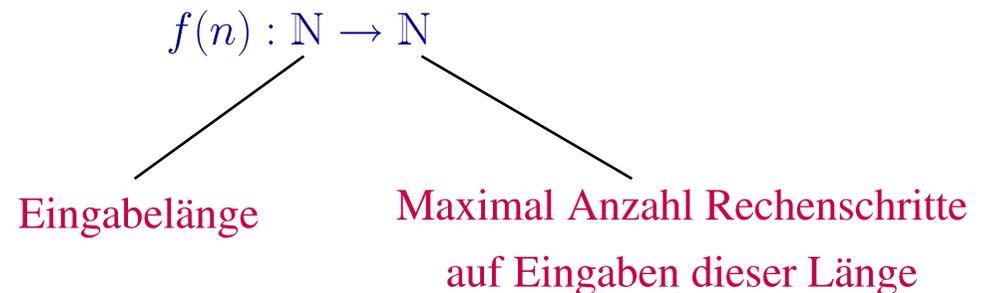


Exkurs Komplexitätsanalyse

Grundideen:

- Laufzeit eines Algorithmus A auf Eingabe x ist die Anzahl **elementarer Rechenschritte**, die A gestartet auf x ausführt (Zuweisungen, Additionen, Multiplikationen, etc.)
- Man misst die Laufzeit in Abhängigkeit von der **Länge** der Eingabe x , **abstrahiert von konkreter Eingabe**.
- Beschreibung also durch Funktion

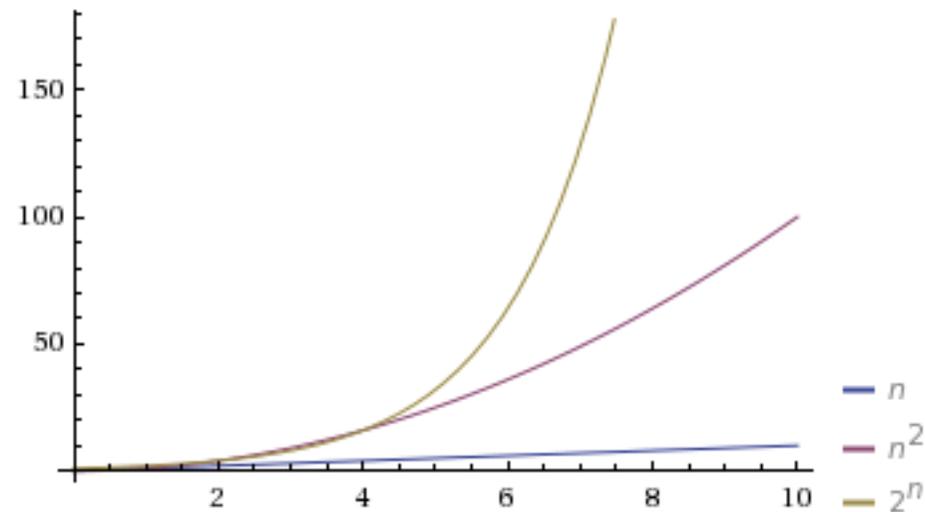


Exkurs Komplexitätsanalyse

Die Grenze zwischen **effizient** und **ineffizient** wird meist angesetzt bei:

- **effizient**: **polynomielle** Laufzeit
Funktion $f(n)$ ist Polynom (beliebigen Grades): n^2 , $5 \cdot n^8$, etc.
- **ineffizient**: **exponentielle** Laufzeit
also Funktionen f der Art 2^n , $6^{3 \cdot n}$, etc.

Besonders gut ist **lineare** Laufzeit $c \cdot n$.



Exkurs \mathcal{O} -Notation

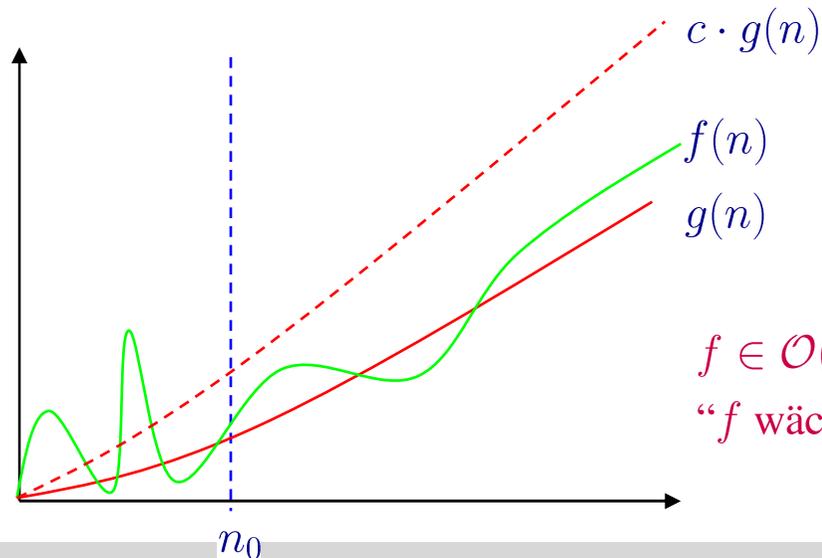
Bei Laufzeit-Analyse möchte man meist **von konkreten Konstanten abstrahieren**
also nicht unterscheiden zwischen z.B. $3 \cdot n^2$ und $5 \cdot n^2$

Definition (\mathcal{O} -Notation)

Seien f und g Funktionen von \mathbb{N} nach \mathbb{N} . Man schreibt

$$f \in \mathcal{O}(g)$$

wenn es $c > 0$ und $n_0 \geq 0$ gibt so dass $f(n) \leq c \cdot g(n)$ für alle $n > n_0$.



$f \in \mathcal{O}(g)$ intuitiv:

“ f wächst nicht wesentlich schneller als g ”



Exkurs \mathcal{O} -Notation

Laufzeit $\mathcal{O}(f(n))$ heisst also intuitiv: **Laufzeit $f(n)$, bis auf Konstanten.**

Insbesondere beschreibt

- $\mathcal{O}(n)$ Linearzeit
- $\mathcal{O}(n^2)$ quadratische Zeit
- $\bigcup_{c \geq 1} \mathcal{O}(n^c)$ polynomielle Zeit

Einfache Rechenregeln z.B.:

- $\mathcal{O}(\mathcal{O}(f(n))) = \mathcal{O}(f(n))$
- $\mathcal{O}(f(n)) + \mathcal{O}(g(n)) = \mathcal{O}(f(n) + g(n))$



Exkurs Entscheidbarkeit

Wir werden später sehen (TheoInf II):

Manche (mathematisch wohl-definierte) Probleme sind nicht berechenbar, für sie gibt es **gar keinen** Algorithmus (egal wie ineffizient)

Beispiel:

gegeben ein Java-Programm, bestimme ob es **auf allen Eingaben anhält oder nicht**

Wichtige Begriffe:

- **Entscheidungsproblem**: Problem, das nur ja/nein-Antworten erfordert
- **Entscheidbares** Problem: Entscheidungsproblem, das algorithmisch berechenbar ist

