8.6 Der Laufzeitbeschleunigungssatz

Oft hilft bei der Laufzeitanalyse rekursiver Funktionen folgender Satz:

Satz 8.1. *Es sei $f : \mathbb{N} \to \mathbb{N}$ eine monoton wachsende Funktion, $a \in \mathbb{N}_{>1}, b, c \in \mathbb{R}_{>0}$, so dass $f(1) \leq \frac{c}{2}$ und die folgende Ungleichung gilt:*

$$f(a \cdot n) \leq b \cdot f(n) + c \cdot n \text{ für alle } n \geq 1.$$

Dann gilt:

$$f(n) \in \begin{cases} \mathcal{O}(n) & \text{falls} \quad a > b \\ \mathcal{O}(n \log n) & \text{falls} \quad a = b \\ \mathcal{O}(n^{\log_2 b}) & \text{falls} \quad a < b \end{cases}$$

Beispiel.
- Karazubas Multiplikation: $a = 2, b = 3 \implies \in \mathcal{O}(n^{\log_2 3})$
- MergeSort: $a = 2, b = 2 \implies \in \mathcal{O}(n \log n)$

Beweis. Zeige zunächst:

$$f(a^q) \leq c \cdot a^{q-1} \cdot \sum_{i=0}^{q} \left(\frac{b}{a}\right)^i$$

Induktionsanfang ($q = 0$):

$$\begin{aligned} f(a^0) = f(1) &\overset{!}{\leq} \frac{c}{a} \leq c \cdot a^{0-1} \cdot \sum_{i=0}^{0} \left(\frac{b}{a}\right)^i \\ &\leq \frac{c}{a} \end{aligned}$$

Induktionsschluss ($q \to q + 1$):

$$\begin{aligned} f(a^{q+1}) &= f(a \cdot a^q) \\ &\leq b \cdot f(a^q) + c \cdot a^q \\ &\leq b \cdot c \cdot a^{q-1} \cdot \sum_{i=0}^{q} \left(\frac{b}{a}\right)^i + c \cdot a^q \\ &= c \cdot a^q \cdot \sum_{i=0}^{q} \left(\frac{b}{a}\right)^{i+1} + c \cdot a^q \\ &= c \cdot a^q \left(\sum_{i=1}^{q+1} \left(\frac{b}{a}\right)^i + 1 \right) \\ &= c \cdot a^q \sum_{i=0}^{q+1} \left(\frac{b}{a}\right)^i \end{aligned}$$

Wir haben gezeigt:

$$f(a^q) \leq c \cdot a^{q-1} \cdot \sum_{i=0}^{q} \left(\frac{b}{a}\right)^i \text{ für alle } q \in \mathbb{N}$$

Folglich gilt:

$$f(n) \leq f(a^{\lceil \log_a n \rceil}) \leq c \cdot n \cdot \sum_{i=0}^{\lceil \log_a n \rceil} \left(\frac{b}{a}\right)^i \text{ für alle } n \in \mathbb{N}_{>0}$$

1. Fall, $a > b$

$$\begin{aligned} f(n) &\leq c \cdot n \cdot \sum_{i=0}^{\lceil \log_a n \rceil} \Big(\frac{b}{a}\Big)^i \\ &\leq c \cdot n \cdot \sum_{i=0}^{\infty} \Big(\frac{b}{a}\Big)^i \\ &= \frac{c \cdot n}{1 - \frac{b}{a}} \in \mathcal{O}(n) \end{aligned}$$

2. Fall, $a = b$

$$\begin{aligned} f(n) &\leq c \cdot n \cdot \sum_{i=0}^{\lceil \log_a n \rceil} 1 \\ &\leq c \cdot n \cdot (\log_a n + 2) \in \mathcal{O}(n \log n) \end{aligned}$$

3. Fall, $a < b$

$$\begin{aligned} f(n) &\leq c \cdot n \cdot \sum_{i=0}^{\lceil \log_a n \rceil} \Big(\frac{b}{a}\Big)^i \\ &= c \cdot n \cdot \frac{\big(\frac{b}{a}\big)^{\lceil \log_a n \rceil + 1} - 1}{\frac{b}{a} - 1} \\ &\leq \frac{c}{\frac{b}{a} - 1} \cdot n \cdot \Big(\frac{b}{a}\Big)^{\log_a n + 2} \\ &= \frac{c}{\frac{b}{a} - 1} \cdot \Big(\frac{b}{a}\Big)^2 \cdot n \cdot \frac{b^{\log_a n}}{a^{\log_a n}} \\ &= \frac{c}{\frac{b}{a} - 1} \cdot \Big(\frac{b}{a}\Big)^2 \cdot b^{\log_a n} \in \mathcal{O}(n^{\log_a b}) \end{aligned}$$

□

8.7 Untere Schranke für Sortieren

Satz 8.2. *Jeder auf paarweisen Vergleichen basierende Algorithmus benötigt zum Sortieren einer n-elementigen Menge im Worst-Case $\Omega(n \log n)$ Vergleiche.*

Bemerkung.

- Der Satz gilt zunächst nur für deterministische (im Vergleich zu randomisierten) Algorithmen.
- Man kann sogar zeigen, dass die untere Schranke nicht nur im Worst-Case, sondern sogar im Mittel gilt.
- MergeSort ist also mit Blick auf die asymptotische Laufzeit ein bestmöglicher vergleichsbasierter Sortieralgorithmus.

- Es gibt Sortieralgorithmen, die nicht vergleichsbasiert sind und bessere Laufzeit erzielen (bspw. BucketSort, RadixSort).