2.3 Sortieren

2.3.1 Einleitung
2.3.2 Einfache Sortierverfahren
2.3.3 Höhere Sortierverfahren
2.3.4 Komplexität von Sortierverfahren
2.3.5 Spezielle Sortierverfahren

Fragestellung: Wie komplex ist Sortieren überhaupt? Gibt es eine Komplexitätsschranke?

Optimaler Sortieralgorithmus

- Was ist die theoretische Untergrenze für die Komplexität eines Sortieralgorithmus?
- Wann können wir mit einem Sortieralgorithmus zufrieden sein?

Optimaler Sortieralgorithmus

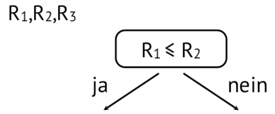
- Reprise: Sortierproblem
  - gegeben:
    - Objektmenge R
    - Ordnung $\leqslant$ auf R^2
    - $R_1, R_2, \ldots, R_n$
  - gesucht: Permutation π mit
    $R_{\pi(1)} \leqslant R_{\pi(2)} \leqslant \ldots \leqslant R_{\pi(n)}$
- Sogenanntes “Sortieren durch Vergleichen”

Jede mögliche Permutation muss möglich sein, da ein Sortieralgorithmus für jede beliebige Permutation der Eingabedaten die richtige Sortierfolge ausgeben soll.

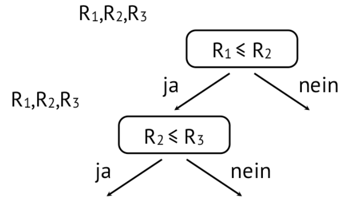
Optimaler Sortieralgorithmus

- Der Ablauf eines nur auf Vergleichen basierenden Sortieralgorithmus kann durch einen Entscheidungsbaum dargestellt werden.
  - innere Knoten = Vergleich
  - Blätter = sortierende Permutation

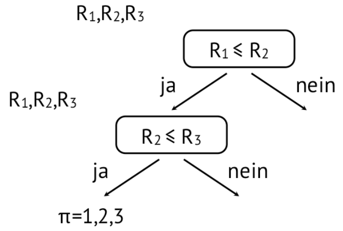
Optimaler Sortieralgorithmus



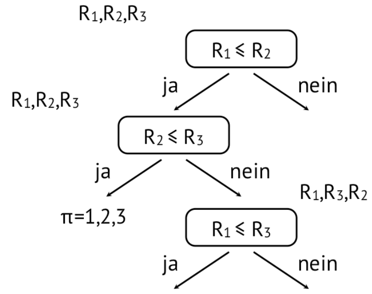
Optimaler Sortieralgorithmus



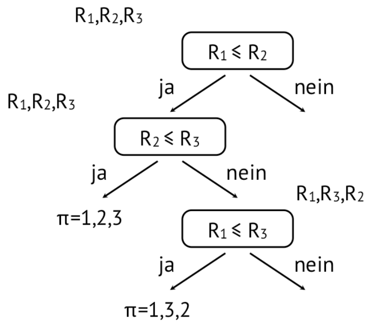
Optimaler Sortieralgorithmus



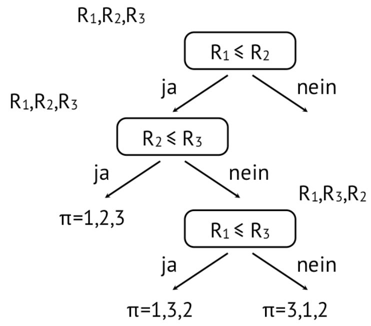
Optimaler Sortieralgorithmus

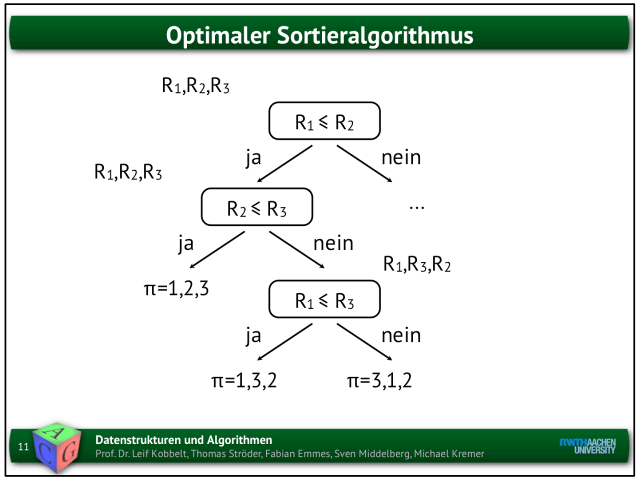


Optimaler Sortieralgorithmus

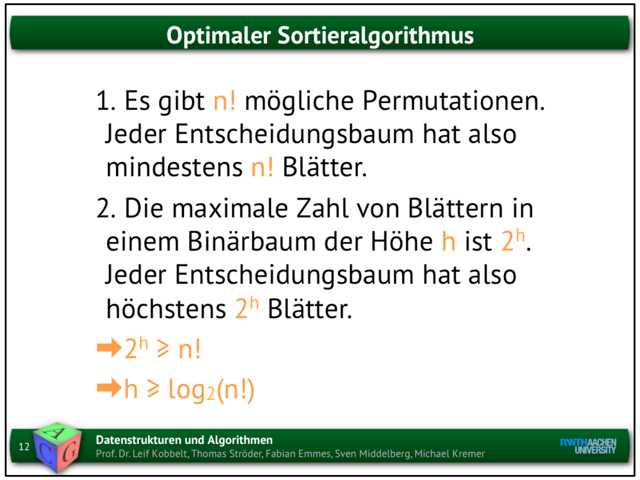


Optimaler Sortieralgorithmus





Jeder Sortieralgorithmus muss jede dieser Permutationen erzeugen können. Dieser Baum hat n! viele Knoten. Die verschiedenen Sortieralgorithmen unterscheiden sich ggf. in der Reihenfolge,
in der die Knoten/der Baum traversiert wird.



Die Anzahl der Blattknoten muss also mindestens n! sein, damit alle mögichen Permutationen im Baum vertreten sind.

Optimaler Sortieralgorithmus

Möglichkeit 1

$$\begin{aligned} h &\geq \log_2(n!) \\ &= \log_2 \prod_{i=1}^{n} i \\ &= \sum_{i=1}^{n} \log_2 i \\ &\geq \sum_{i=1}^{\lceil n/2 \rceil - 1} \log_2 i + \sum_{i=\lceil n/2 \rceil}^{n} \log_2 i \\ &\geq 0 + \lceil n/2 \rceil \log_2 \lceil n/2 \rceil \\ &\geq \frac{n}{2} \log_2 \frac{n}{2} = \frac{n}{2}(\log_2 n - \log_2 2) = \Theta(n \log n) \end{aligned}$$

n log n ist untere Schranke für die Komplexität von Sortieralgorithmen (gezeigt anhand vom Entscheidungsbaum).

Optimaler Sortieralgorithmus

Möglichkeit 2:
Integralmethode

$$\begin{aligned} h &\geq \log_2(n!) \\ &= \sum_{i=1}^{n} \log_2(i) \\ &= c \sum_{i=1}^{n} \ln(i) \\ &\geq c \int_{1/2}^{n+1/2} \ln(x)dx \\ &= c \left(x \ln(x) - x \Big|_{1/2}^{n+1/2} \right) \\ &= \Omega(n \log(n)) \end{aligned}$$

Eine andere Art und Weise, diese untere Schranke zu beweisen.