

KAPITEL 5: SORTIEREN

5.1 Grundlagen: Sortieren mit Vergleichen

- Vorspann:
- Sortierbeispiel (Demo)
 - Diskussion Grundlagen, Unterschiede von Modellen und wesentliche Elemente (insbesondere: Sortieren auf der Basis von Vergleichen!)

Problem 5.1

Gegeben: Eine Menge von n Objekten x_1, \dots, x_n , eine Größenrelation " $<$ ", die je zwei Objekte ordnet (\rightarrow Totalordnung!)

Gesucht: Eine Sortierung der Objekte nach Reihenfolge

Wichtig: Wir müssen die Objekte nicht explizit (physisch) umsortieren - es reicht, die relative Anordnung zu kennen.

Definition 5.2 (Permutation)

Eine Permutation π ist eine Umsortierung von n Objekten, d.h. eine bijektive Abbildung

$$\begin{aligned} \pi: \{1, \dots, n\} &\rightarrow \{1, \dots, n\} \\ i &\mapsto \pi(i). \end{aligned}$$

Man schreibt π auch in folgender Weise:

$$\begin{pmatrix} 1 & \dots & n \\ \pi(1) & \dots & \pi(n) \end{pmatrix}$$

- d.h. als Wertetabelle.

Beobachtung 5.3

Es gibt genau $n!$ Permutationen von n Objekten.

Wie lange benötigt man zum Sortieren mithilfe von nicht mehr als paarweisem Vergleichen?

Satz 5.4

Für n Objekte x_1, \dots, x_n benötigt man zum Sortieren mindestens $\Omega(n \log n)$, wenn man die Objekte nur paarweise vergleichen kann.

Beweis:

- (1) Zunächst hat man $n! = n(n-1) \dots 2 \cdot 1$ viele mögliche Permutationen.
- (2) Jeder Vergleich teilt die Menge der verbleibenden möglichen Permutationen in zwei Teilmengen.
- (3) Im schlechtesten Falle bleibt jeweils die größere Teilmenge übrig.
- (4) Man kann also bestenfalls eine Halbierung der verbleibenden Menge von Permutationen erzwingen.
- (5) Bis man eine eindeutige Permutation sicher identifiziert hat, braucht man also mindestens $\log_2(n!)$ Vergleiche.

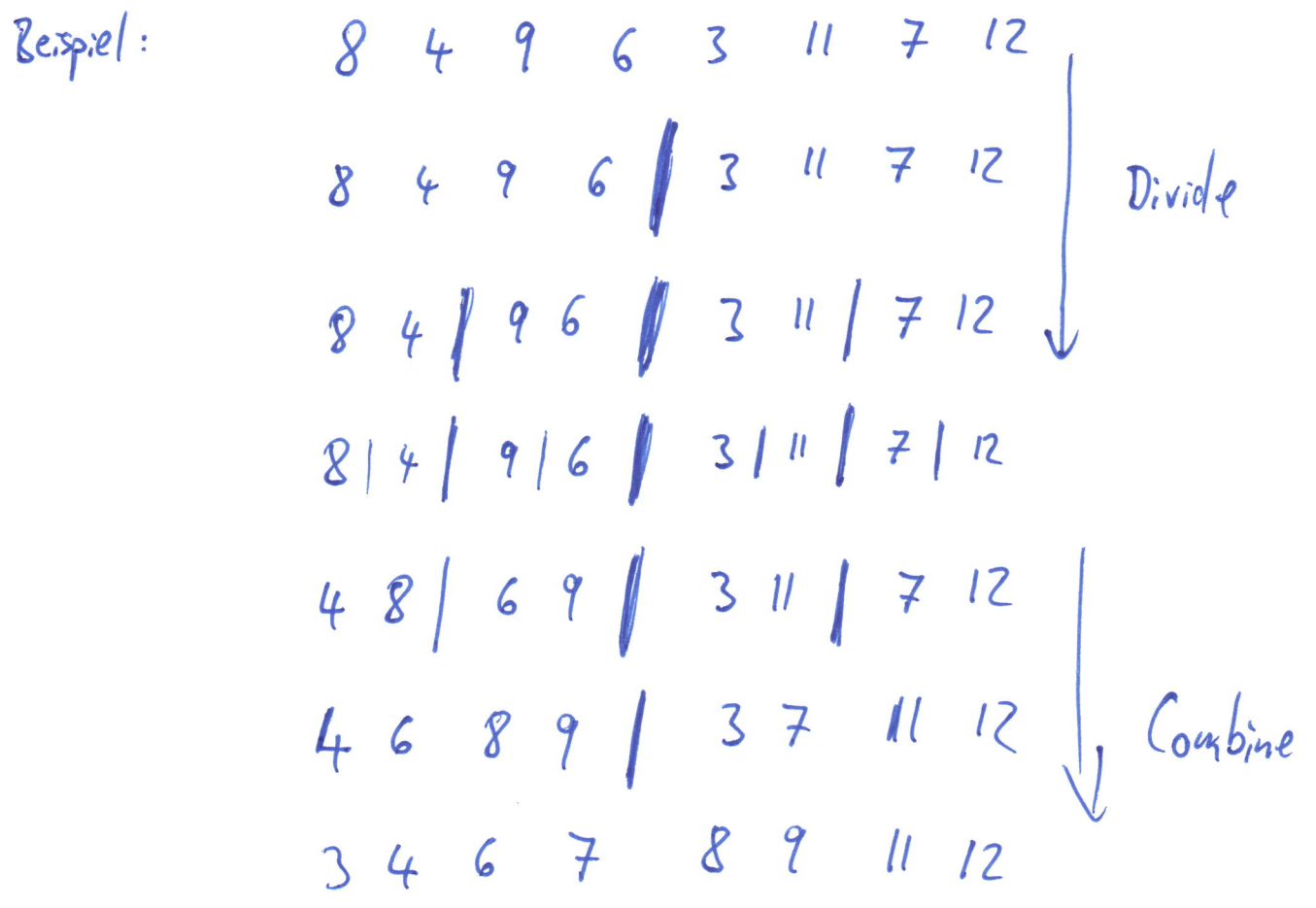
$$\begin{aligned}
 (6) \quad \log_2(n!) &= \sum_{i=1}^n \log_2 i \\
 &\geq \sum_{i=\frac{n}{2}}^n \log_2 i \\
 &\geq \frac{n}{2} \log_2 \frac{n}{2} \\
 &= \frac{n}{2} (\log_2 n - 1) \\
 &\in \Omega(n \log n)
 \end{aligned}$$



5.2 Mergesort

Idee: "Divide and Conquer", d.h. "Teile und herrsche"
(DIVIDE ET IMPERA)

Genauer: Divide, conquer, combine!
-> Teile, herrsche, füge zusammen!



Laufzeit: $\log n$ Zeilen zum Unterteilen
 $\log n$ Zeilen zum Zusammenfügen
 Jeweils n Schritte pro Zeile beim Zusammenfügen!