

# Theo 3

Nick Daiber

October 14, 2024

## 1 Sortieren

Problem: Gegeben eine  $n$ -elementige Teilmenge  $S = \{S_1, \dots, S_n\}$  aus geordneten Universum  $U$ .

Gesucht: Permutation  $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ , sodass gilt  $S_{\pi(1)} < S_{\pi(2)} < S_{\pi(3)} < \dots$

### 1.1 Ordnung

- Buchstaben nach Position im Alphabet
- $w_1 < w_2 \Leftrightarrow w_1 = s_1 x t_1, w_2 = s_1 y t_2$   
mit  $s_1 \in a, \dots, Z^*$

Sortierverfahren berechnet  $\pi$  Aufwand:

- Zeit
- Platz
- Anzahl der Zufallsbits
- I/O operationen

Analyse eines Algorithmus für beliebige Probleminstanzen oder mit speziellen Eigenschaften.

Wir betrachten nur beliebige Probleminstanzen.

### 1.2 Bubble sort

Eingabe: Array  $A[1 \dots n]$  (Zahlen)

Ausgabe:  $A[]$  so sortiert, dass  $\forall 1 \leq i \leq n-1 : A[i] \leq A[i+1]$

### 1.2.1 Sortiert Bubblesort A?

- Alg verliert keine Zahl (weiß nicht, ob das wirklich bewiesen ist)

Lemma: Sei  $i$  beliebig, betrachte  $A[1, i]$  zu Zeitpunkt 1, 2 zu 2 gilt  $A[i] = \max(g \in 1, \dots, i) A(g) \Rightarrow$  das größte Element steht in  $A[i]$  und kein Element geht verloren

Beweis: Idee - Sei  $h$  der index in  $1, \dots, i$  mit  $A[h] = \max$  sobald  $j = h$  wird  $A[h]$  mit  $A[h + 1]$  getauscht, etc

## 1.3 Laufzeit

Uns interessiert die Laufzeit von BS

- Implementiere Alg und messen  
Problem: Andere Hardware, etc.
- Anzahl der Instruktionen zählen (im Code)
- Anzahl Vergleiche Zählen

Behauptung: Abgesehen von Konstanten Faktor weicht das nicht ab vom Zählen aller ausgeführten Instruktionen

Annahme: Algorithmen haben konstante Größe

Lemma: Wenn eine Algorithmenbeschreibung  $c$  Zeilen hat, wird bei erfolgreichem Ablauf mindestens alle  $c$  Zeilen ein Vergleich ausgeführt

$$\text{Vergleiche} = n + \sum_{l=2}^n l + \sum_{l=2}^{n-1} l = n^2 + n - 2$$

Bubblesort macht  $n^2 + n - 1$  vgl.

Annahme: Rechner macht  $10^9$  instruktionen pro sekunde