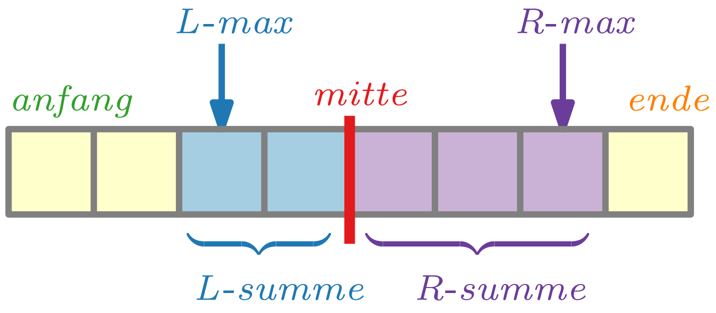
Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

# Algorithmen und Datenstrukturen

## Vorlesung 4: Maximales Teilfeld

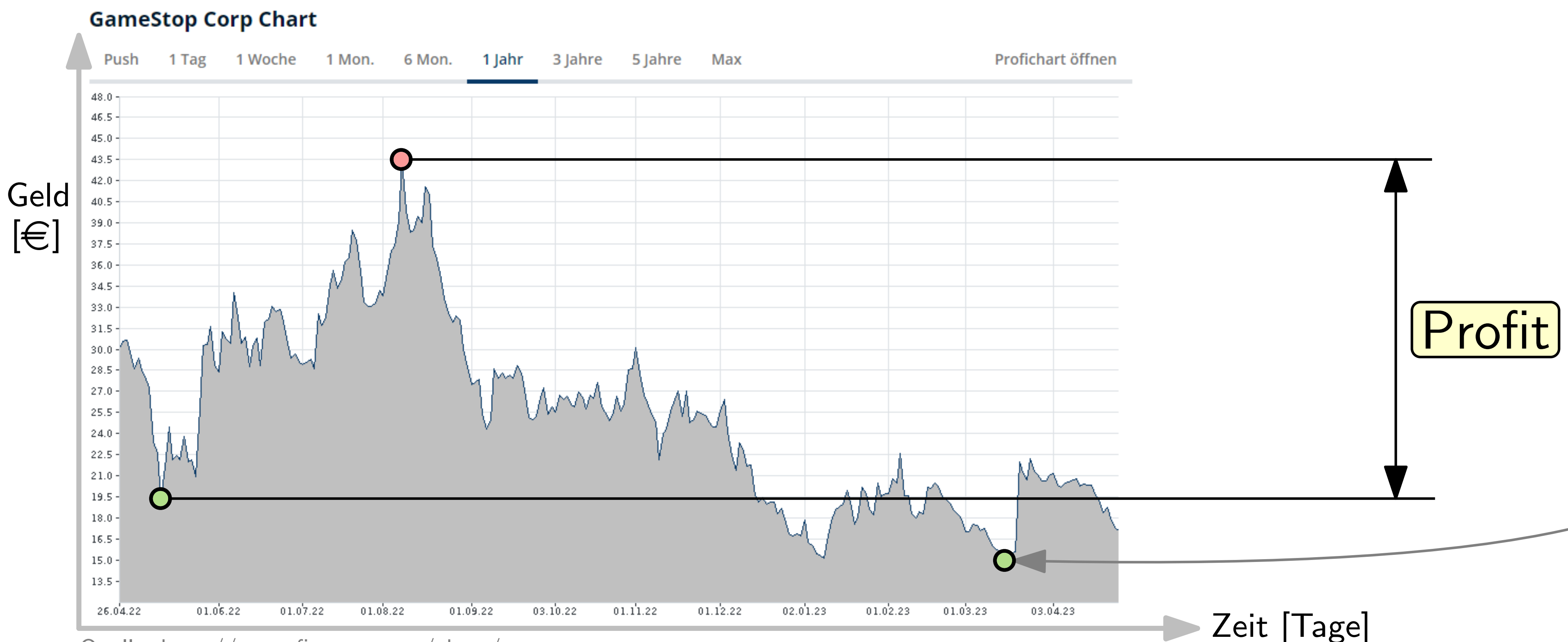




*Alexander Wolff*

*Wintersemester 2025*

# Analyse von Aktienkursen




Quelle: http://www.finanzen.net/chart/gamestop

**Wichtig:**
Es genügt **nicht** Minimum und Maximum zu suchen!

**Problem.** **Gegeben:** Eine Folge $A[1 \dots n]$ von Aktienkursen in Euro.

**Gesucht:** Paar $(i, j)$ mit $1 \leq i < j \leq n$, so dass $\underbrace{\underbrace{A[j]}_{\text{Verkaufskurs}} - \underbrace{A[i]}_{\text{Einkaufskurs}}}_{\text{Profit pro Aktie}}$ **maximal**.

# Analyse von Aktienkursen

**Problem.** MaxDiff

**Gegeben:** Eine Folge $A[1\ldots n]$ von Aktienkursen in Euro.

**Gesucht:** Paar $(i,j)$ mit $1 \le i < j \le n$, so dass $A[j] - A[i]$ **maximal**.

**Lösung:** per **roher Gewalt**

**Übung.** Schreiben Sie Pseudocode!

- für alle erlaubten Paare $(i,j)$ berechne $A[j] - A[i]$
- gib Maximum zurück

**Laufzeit.** $\approx$ Anzahl der berechneten Differenzen

$=$ Anzahl erlaubter Paare

$$= (n-1) + (n-2) + \ldots + 2 + 1 \;=\; \frac{n(n-1)}{2} \;\in\; \Theta(n^2)$$

1) $\sum_{i=1}^{n} i \;=\; \frac{n(n+1)}{2}$ **arithmetische Reihe**

# Ein ähnliches Problem

**Problem.** MaxSum

**Gegeben:** Eine Folge $A[1 \dots n]$ von ganzen Zahlen.

**Gesucht:** Paar $(i, j)$ mit $1 \leq i \leq j \leq n$, so dass $\sum_{k=i}^{j} A[k]$ **maximal**.

| 7 | 4 | -9 | 1 | -3 | 3 | 1 | 12 | 0 | -2 | 0 | 4 | 2 | -8 | 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

$\Rightarrow (6, 13)$ oder $(1, 13)$

**Lösung:** per **roher Gewalt**

**Übung.** Schreiben Sie Pseudocode!

- für alle erlaubten Paare $(i, j)$ berechne $\sum_{k=i}^{j} A[k]$
- gib Maximum zurück

**Laufzeit.** $\approx$ Anzahl der Additionen

Obere Schranke dafür: $\mathcal{O}(n^2) \cdot \mathcal{O}(n) = \mathcal{O}(n^3)$

Untere Schranke (Anz. Paare) $= \Omega(n^2)$

**Wo ist die Wahrheit?**

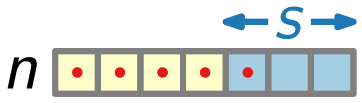
# Genauere Analyse

**Laufzeit.** $\approx$ Anzahl der Additionen des Rohe-Gewalt-Algos:

- für alle erlaubten Paare $(i, j)$ berechne $\sum_{k=i}^{j} A[k]$
- gib Maximum zurück

**Beobachtung.**

- Anz. der Summen mit $s$ Summanden ist $n - s + 1$.
- $s$ Summanden benötigen $s - 1$ Additionen.




$\Rightarrow$ Anz. Add. $= \sum_{s=1}^{n}(n - s + 1) \cdot (s - 1)$

$= n \cdot 0 + (n-1) \cdot 1 + (n-2) \cdot 2 + \ldots + 2 \cdot (n-2) + 1 \cdot (n-1)$

$\underset{(\text{falls } 4 \mid n)}{=} \ldots + \underbrace{\overbrace{\frac{3n}{4} \cdot \frac{n}{4} + \ldots + \frac{n}{2} \cdot \frac{n}{2} + \ldots + \frac{n}{4} \cdot \frac{3n}{4}}}_{\frac{n}{2} + 1 \text{ Terme der Größe mindestens } \frac{n}{4} \cdot \frac{n}{4}} + \ldots \in \Omega(n^3)$

**Übung.**
Berechnen Sie diese Summe **genau** und beweisen Sie Ihr Ergebnis per Induktion!

**Wie berechnen?**
$\#$ Add. $= an^3 + bn^2 + cn + d$
Wertetabelle für $n = 1,2,3,4$.
LGS aufstellen + lösen!

$\Rightarrow$ Der Rohe-Gewalt-Alg. läuft in $\mathcal{O}(n^3) \cap \Omega(n^3) = \Theta(n^3)$ Zeit.

**Geht das besser?**

# Eine schnellere Lösung

**Problem.** MaxSum

**Gegeben:** Eine Folge $A[1 \ldots n]$ von ganzen Zahlen.

**Gesucht:** Paar $(i, j)$ mit $1 \le i \le j \le n$, so dass $S_{ij} = \sum_{k=i}^{j} A[k]$ **maximal**.

**Idee:** Für $i = 1, \ldots, n$

berechne $S_{ii}, \quad S_{i,i+1}, \quad S_{i,i+2}, \quad S_{i,i+3}, \quad \ldots, \quad S_{i,n}$

$=\quad +\quad +\quad +\quad \ldots\quad +$

Wie? $\quad A[i] \quad \underbrace{A[i+1] \quad A[i+2] \quad A[i+3] \quad \ldots \quad A[n]}_{}$

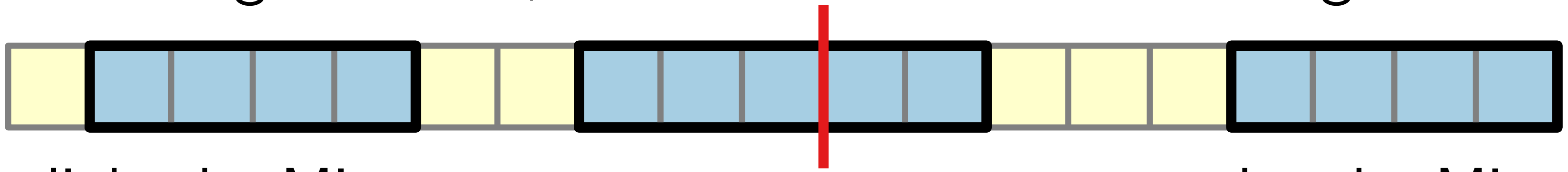
$n - i$ Additionen

Insgesamt $\displaystyle\sum_{i=1}^{n} n - i \;= \sum_{j=n-1}^{0} j \;= \sum_{j=1}^{n-1} j \in \Theta(n^2)$ Additionen

1) $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ **arithmetische Reihe**

# Eine noch schnellere Lösung?

**Idee:** Drei Möglichkeiten, wo maximale Teilsumme liegt:




Nimm Entwurfstechnik **Teile & Herrsche**!

- **teile:** in zwei ungefähr gleichgroße Hälften
- **herrsche:** durch rekursive Aufrufe für linke und rechte Hälfte
- **kombiniere:** kontrolliere alle Teilsummen, die die Mitte enthalten

Davon gibt's $\frac{n}{2} \cdot \frac{n}{2} \in \Theta(n^2)$ ☹

**Einsicht:** Wenn die **maximale** Teilsumme die Mitte enthält,
dann muss ihr linker Teil (bis zur Mitte) maximal sein
**und** dann muss ihr rechter Teil (ab der Mitte) maximal sein.

$\Rightarrow$ Können linken und rechten Teil **unabhängig** voneinander berechnen!

# Teile & Herrsche

$\textsc{MaxTeilfeld}(\text{int}[]\ A, \text{int}\ anfang = 1, \text{int}\ ende = A.length)$

**if** $anfang == ende$ **then**
  **return** $(anfang, ende, A[anfang])$ } **herrsche** (in kleinen Teilinstanzen)

**else**

  $mitte = \lfloor (anfang + ende)/2 \rfloor$ } **teile**

  $(L\text{-}anfang, L\text{-}ende, L\text{-}summe) = \textsc{MaxTeilfeld}(A, anfang, mitte)$

  $(R\text{-}anfang, R\text{-}ende, R\text{-}summe) = \textsc{MaxTeilfeld}(A, mitte+1, ende)$ } **herrsche**

  $(M\text{-}anfang, M\text{-}ende, M\text{-}summe) = \textsc{MaxMittTeilfeld}(A, anfang, mitte, ende)$ } **kombiniere**

  **return** (Tripel mit größter Summe)

**Laufzeit:** $T_{\mathrm{MT}}(1) = \Theta(1)$

für $n > 1$: $T_{\mathrm{MT}}(n) = T_{\mathrm{MT}}(\lfloor n/2 \rfloor) + T_{\mathrm{MT}}(\lceil n/2 \rceil) + T_{\mathrm{MMT}}(n)$

$$\approx 2 \cdot T_{\mathrm{MT}}(n/2) + T_{\mathrm{MMT}}(n)$$

$$T_{\mathrm{MMT}}(n) = ?$$

# Kombiniere

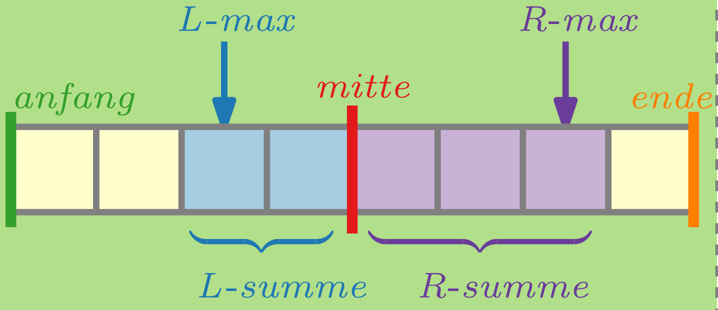
MaxMittTeilfeld(int[] $A$, int $anfang$, int $mitte$, int $ende$)

$L\text{-}summe = -\infty$

$summe = 0$

**for** $i = mitte$ **downto** $anfang$ **do**

Vervollständigen Sie den Algorithmus!




$R\text{-}summe = -\infty$

$summe = 0$

**for** $i = mitte + 1$ **to** $ende$ **do**

// analog zu oben

**return** $(L\text{-}max, R\text{-}max, L\text{-}summe + R\text{-}summe)$

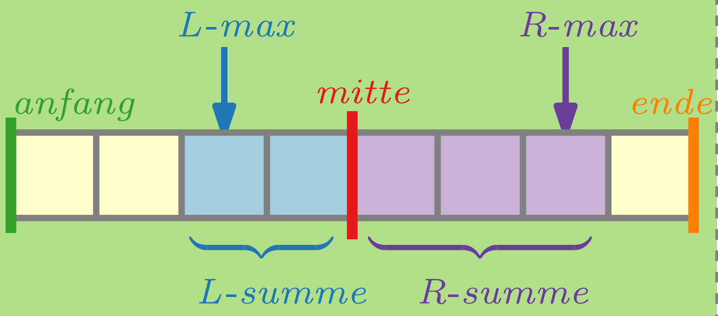
# Kombiniere

$\textsc{MaxMittTeilfeld}$(int[] $A$, int $anfang$, int $mitte$, int $ende$)

```
L-summe = −∞
summe = 0
for i = mitte downto anfang do
    summe = summe + A[i]
    if summe > L-summe then
        L-summe = summe
        L-max = i

R-summe = −∞
summe = 0
for i = mitte + 1 to ende do
    // analog zu oben   +

return (L-max, R-max, L-summe + R-summe)
```




**Korrektheit?** ✓

Schleifeninvariante:

$summe = S_{i,mitte}$ und

$L\text{-}summe = \max_{i \le k \le mitte} S_{k,mitte}$

**Laufzeit?** ✓

$:=_{\text{hier}}$ Anz. Additionen

$= mitte - anfang + 1$

$+ ende - mitte$

$= ende - anfang + 1$

$= n$

# Putting Things Together

## Laufzeit von MaxTeilfeld:

$$T_{\mathrm{MT}}(1) = \Theta(1)$$

für $n > 1$:

$$T_{\mathrm{MT}}(n) \approx 2 \cdot T_{\mathrm{MT}}(n/2) + T_{\mathrm{MMT}}(n)$$
$$= 2 \cdot T_{\mathrm{MT}}(n/2) + n$$
$$= V_{\mathrm{MS}}(n) \quad = \mathcal{O}(n \log_{\cancel{2}} n)$$

~~[für $n$ = Zweierpotenz]~~

„Runde auf" zu nächster Zweierpotenz $n' < 2n$

MergeSort

Denn für $a, b \geq 2$ gilt: $\Theta(\log_a n) = \Theta(\log_b n)$.

## Denkaufgaben:

- Lösen Sie MaxSum in $\mathcal{O}(n)$ – also in linearer – Zeit!
- **Und wenn...** $\quad T(n) = 2 \cdot T(n/2) + 4n \quad$ (und $T(1) = \Theta(1)$)

  Gilt dann auch $\quad T(n) = \mathcal{O}(n \log n)$ **?**

# Übersicht

*) Wir setzen die Konstante $c$ in der $\mathcal{O}$-Notation auf 1.

- Anzahl der Additionen
- geschätzte Rechenzeit

für einen Kern (CPU) à 1 GHz, d.h. $10^9$ Add./s

| **Algorithmus** | **Laufzeit** | $n = 1\,000$ | | $n = 1\,000\,000$ | |
|---|---|---|---|---|---|
| Rohe Gewalt | $\mathcal{O}(n^3)$ | $10^9$ | 1 s | $10^{18}$ | 31,7 y |
| Reihenfolge der Summen ändern | $\mathcal{O}(n^2)$ | $10^6$ | 1 ms | $10^{12}$ | 1000 s $= 17$ min |
| Teile & Herrsche | $\mathcal{O}(n \log_{10} n)$ | $3 \cdot 10^3$ | $3\,\mu$s | $6 \cdot 10^6$ | 6 ms |
| Linearer Scan (siehe Buch [CLRS], Ü-Aufgabe 4.1-5) | $\mathcal{O}(n)$ | $10^3$ | $1\,\mu$s | $10^6$ | 1 ms |