





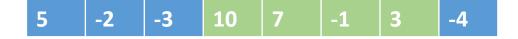
# Grundzüge der Informatik 1

Vorlesung 7 - flipped classroom

## Wiederholung

 Beim Problem maximale Teilsumme soll ein (nicht leerer) zusammenhängender Bereich in einem Feld A gefunden werden, der die Summe seiner Elemente maximiert

# **Beispiel**





## Aufgabe 1

- Entwickeln Sie einen Teile&Herrsche Algorithmus für das Teilsummenproblem
- Es genügt, wenn Sie den Wert der optimalen Lösung bestimmen
- Stellen Sie eine Laufzeitrekursion auf. Können Sie diese Auflösen?



### Teilsumme(A,I,r)

- if l=r then return A[l]
- 2.  $m = \lfloor (l+r)/2 \rfloor$
- 3. mitte = Left(A,I,m) + Right(A,m+1,r)
- 4. **return** max(Teilsumme(A,I,m), Teilsumme(A,m+1,r), mitte)

# Erklärung

- Der Algorithmus soll die maximale Teilsumme in A[l...r] berechnen
- Diese ist entweder in A[I..m] oder in A[m+1...r] oder sie schneidet die Mitte
- Dabei soll die Funktion Left(A,I,m) die größte Teilsumme in A[I...m] finden, die bei m endet und Right(A,m+1,r) die größte Teilsumme aus A[m+1...r], die bei m+1 beginnt. Die Summe ist die größte Teilsumme, die die Mitte schneidet.

Universit

## Left(A,I,m)

- 1. W=0
- 2. max=A[m]
- 3. for i=m downto | do
- 4. W=W+A[i]
- 5. if W> max then max =W
- 6. **return** max



### Laufzeit

- Die Laufzeit von Left(A,I,m) ist O(N), wobei N=m+1-I die Größe des betrachteten Teilfeldes ist
- Gleiches gilt f
  ür Right(A,m+1,r)
- Damit ergibt f
  ür Teilsumme(A,I,r) eine Laufzeit von
- T(n) = 2 T(n/2) + O(n), wobei n jeweils der Anzahl der Elemente im betrachteten Teilfeld entspricht, sowie T(1) = O(1)
- Dies entspricht der Laufzeitrekursion von Mergesort und ergibt T(n)= O(n log n)



# Laufzeitanalyse - Wo ist der Fehler?

## **Aufgabe 2: Falsche Behauptung**

Mergesort hat Laufzeit O(n).

### **Falscher Beweis**

- Wir zeigen die Aussage nur für den Fall, dass n eine Zweierpotenz ist.
- Die Laufzeit für T(2) ist O(1). Wir zeigen per Induktion, T(n)=O(n) für alle n≥2.
- Induktionsanfang: für n=2 gilt T(2) = O(1).
- Induktionsannahme: Für Eingabelänge m<n, m Zweierpotenz, ist die Laufzeit T(m) = O(m).
- Induktionsschluss: Sei n eine Zweierpotenz. Es gilt T(n) = 2 T(n/2) + O(n).
   Nach Induktionsannahme gilt T(n) = 2 O(n) + O(n) = O(n)
- Also gilt T(n) = O(n).



# Laufzeitanalyse

- Wo ist der Fehler?

# **Aufgabe 2: Falsche Behauptung**

Mergesort hat Laufzeit O(n).

### Wo liegt der Fehler?

- A) Im Induktionsanfang
- B) In der Induktionsannahme
- C) Im Induktionsschluss
- D) Der Beweis ist korrekt. Die Behauptung stimmt nicht, wenn n keine Zweierpotenz ist.

### **Falscher Beweis**

- Wir zeigen die Aussage nur für den Fall, dass n eine Zweierpotenz ist.
- Die Laufzeit für T(2) ist O(1). Wir zeigen per Induktion, T(n)=O(n) für alle n≥2.
- Induktionsanfang: für n=2 gilt T(2) = O(1).
- Induktionsannahme: Für Eingabelänge m<n, m Zweierpotenz, ist die Laufzeit T(m) = O(m).
- Induktionsschluss: Sei n eine Zweierpotenz. Es gilt T(n) = 2 T(n/2) + O(n). Nach Induktionsannahme gilt T(n) = 2 O(n) + O(n) = O(n)
- Also gilt T(n) = O(n).



# Laufzeitanalyse

- Wo ist der Fehler?

# **Aufgabe 2: Falsche Behauptung**

Mergesort hat Laufzeit O(n).

### Wo liegt der Fehler?

- A) Im Induktionsanfang
- B) In der Induktionsannahme
- C) Im Induktionsschluss
- D) Der Beweis ist korrekt. Die Behauptung stimmt nicht, wenn n keine Zweierpotenz ist.

### **Falscher Beweis**

- Wir zeigen die Aussage nur für den Fall, dass n eine Zweierpotenz ist.
- Die Laufzeit für T(2) ist O(1). Wir zeigen per Induktion, T(n)=O(n) für alle n≥2.
- Induktionsanfang: für n=2 gilt T(2) = O(1).
- Induktionsannahme: Für Eingabelänge m<n, m Zweierpotenz, ist die Laufzeit T(m) = O(m).
- Induktionsschluss: Sei n eine Zweierpotenz. Es gilt T(n) = 2 T(n/2) + O(n). Nach Induktionsannahme gilt T(n) = 2 O(n) + O(n) = O(n)
- Also gilt T(n) = O(n).

