Grundzüge der Informatik 1

Vorlesung 7 - flipped classroom

Teile & Herrsche Verfahren

Wiederholung

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

Beispiel

5	-2	-3	10	7	-1	3	-4

Teile & Herrsche Verfahren

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?

Teile & Herrsche Verfahren

Teilsumme(A,l,r)

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

Erklärung

- Der Algorithmus soll die maximale Teilsumme in A[l…r] berechnen
- Diese ist entweder in A[l..m] oder in A[m+1…r] oder sie schneidet die Mitte
- Dabei soll die Funktion Left(A,l,m) die größte Teilsumme in A[l…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.

Teile & Herrsche Verfahren

Left(A,l,m)

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

Teile & Herrsche Verfahren

Laufzeit

- Die Laufzeit von Left(A,l,m) ist O(N), wobei N=m+1-l die Größe des betrachteten Teilfeldes ist
- Gleiches gilt für Right(A,m+1,r)
- Damit ergibt für Teilsumme(A,l,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?

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.

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?

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.

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 $n \geq 2$.
- 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 $T(n) = O(n)$.