

Grundzüge der Informatik 1

Vorlesung 6 - flipped classroom

Algorithmenentwurf durch Rekursion

Teile & Herrsche Verfahren

- Idee: Teile die Eingabe in mehrere gleich große Teile auf
- Löse das Problem rekursiv auf den einzelnen Teilen
- Füge die Teile zu einer Lösung des Gesamtproblems zusammen
- Beispiel: Sortieren durch Aufteilen in zwei Teile

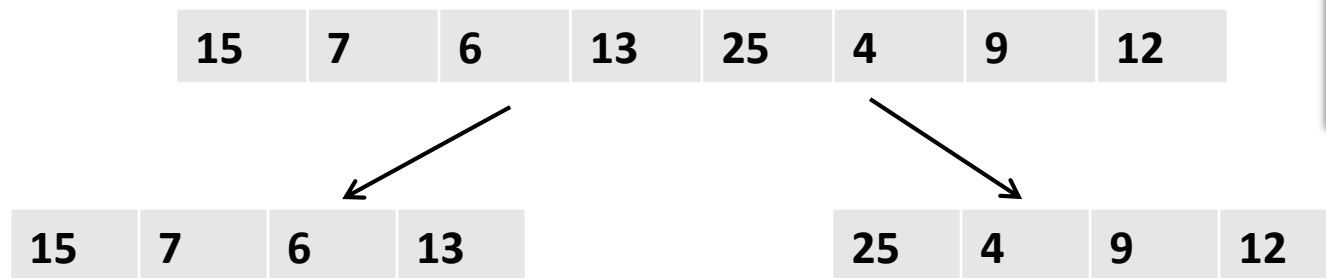
15	7	6	13	25	4	9	12
----	---	---	----	----	---	---	----

Schritt 1:
Aufteilen der
Eingabe

Algorithmenentwurf durch Rekursion

Teile & Herrsche Verfahren

- Idee: Teile die Eingabe in mehrere gleich große Teile auf
- Löse das Problem rekursiv auf den einzelnen Teilen
- Füge die Teile zu einer Lösung des Gesamtproblems zusammen
- Beispiel: Sortieren durch Aufteilen in zwei Teile

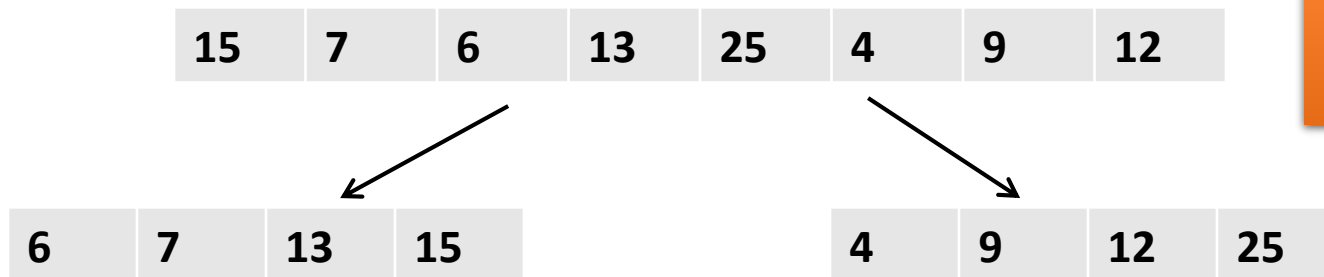


Schritt 1:
Aufteilen der
Eingabe

Algorithmenentwurf durch Rekursion

Teile & Herrsche Verfahren

- Idee: Teile die Eingabe in mehrere gleich große Teile auf
- Löse das Problem rekursiv auf den einzelnen Teilen
- Füge die Teile zu einer Lösung des Gesamtproblems zusammen
- Beispiel: Sortieren durch Aufteilen in zwei Teile



Schritt 2:
Rekursiv Sortieren

Teile & Herrsche Prinzip – Beispiel binäre Suche

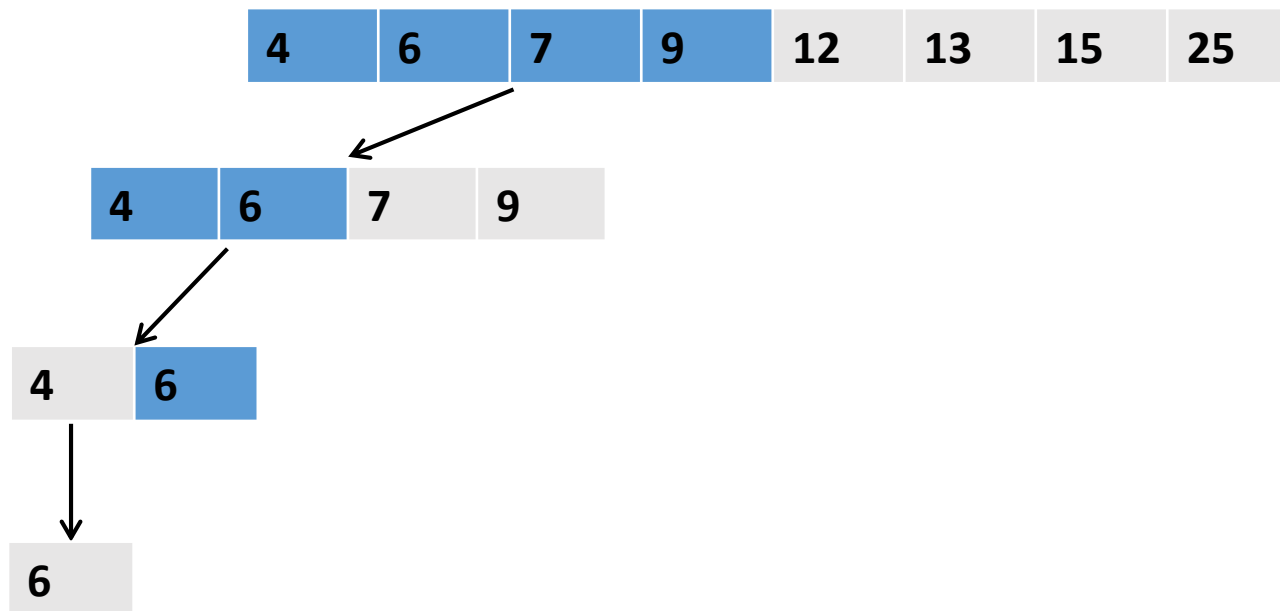
Suche in sortierten Felder

- „Telefonbuchproblem“: Wie können wir in einem sortierte Feld eine bestimmte Zahl finden?
- Die Zahl dient als „Schlüssel“ und mit ihr können weitere Informationen abgespeichert sein
- Verwende Teile & Herrsche Ansatz

Teile & Herrsche Prinzip – Beispiel binäre Suche

Teile & Herrsche Ansatz (hier: Suche nach 6):

- Teile in der Mitte und suche rekursiv, in dem eindeutigen Teil, der das gesuchte Element enthalten kann



Teile & Herrsche Verfahren

Aufgabe 1

- Eine Zirkulation in einem Feld der Größe n ersetzt gleichzeitig jedes Element an Position $i+1$ durch das Element an Position i und das Element an Position 1 durch das an Position n
- Beschreiben Sie die Zirkulation in Pseudocode

Teile & Herrsche Verfahren

Zirkulation(A,n)

1. $\text{last} = A[n]$
2. **for** $i=n-1$ **downto** 1 **do**
3. $A[i+1] = A[i]$
4. $A[1] = \text{last}$

Teile & Herrsche Verfahren

Aufgabe 2

- Sei A ein sortiertes Feld der Länge n auf das eine unbekannte Anzahl Zirkulationen kleiner n angewendet wurden
- Entwickeln Sie einen Teile & Herrsche Algorithmus, der die Anzahl der Zirkulationen bestimmt. Hinweis: Sie benötigen dazu **nicht** den Algorithmus aus Aufgabe 1
- Was ist die Laufzeit Ihres Algorithmus?

Teile & Herrsche Verfahren

AnzahlZirkulationen(A, l, r)

1. **if** $l=r$ **then return** l
2. $m = \lceil (l+r)/2 \rceil$
3. **if** $A[l] > A[m]$ **then return** AnzahlZirkulationen(A, l, m-1)
4. **else return** AnzahlZirkulationen (A m, r)

Idee

- Finde das maximale Element im Feld und gib den Index zurück
- Ist $A[l]$ (linkes Ende) größer als $A[m]$ (Median), dann liegt das Maximum vor dem Median
- Laufzeit wie binäre Suche: $O(\log n)$

Teile & Herrsche Verfahren

Aufgabe 3

- Beim Problem maximale Teilsumme soll ein (nicht leerer) zusammenhängender Bereich in einem Feld A gefunden werden, der die Summe seiner Elemente maximiert
- Entwickeln Sie einen Algorithmus, der das Teilsummenproblem ohne Rekursion löst
- Was ist die Laufzeit Ihres Algorithmus?

Beispiel

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

Teile & Herrsche Verfahren

Teilsumme(A,n)

1. $\text{Max} = -\infty$
2. **for** $i=1$ **to** n **do**
3. **for** $j=i$ **to** n **do**
4. $W=0$
5. **for** $k=i$ **to** j **do**
6. $W=W+A[k]$
7. **if** $W>\text{Max}$ **then** $\text{Max}=W$; $l=i$, $r=j$
8. **return** (Max, l, r)

Teile & Herrsche Verfahren

Teilsumme(A,n)

1. $\text{Max} = -\infty$
2. **for** $i=1$ **to** n **do**
3. **for** $j=i$ **to** n **do**
4. $W=0$
5. **for** $k=i$ **to** j **do**
6. $W=W+A[k]$
7. **if** $W>\text{Max}$ **then** $\text{Max}=W$; $l=i$; $r=j$
8. **return** (Max,l,r)

Laufzeit $O(n^3)$