Universität Karlsruhe (TH)

Informatik II Tutorium

Sommer 2005

Zur Vorlesung von:

Prof. Calmet und
Dipl. Math. Ralf Eberhardt

Tutorium:

Nr. 24

Tutor:

Joachim Wilke

info2@joachim-wilke.de

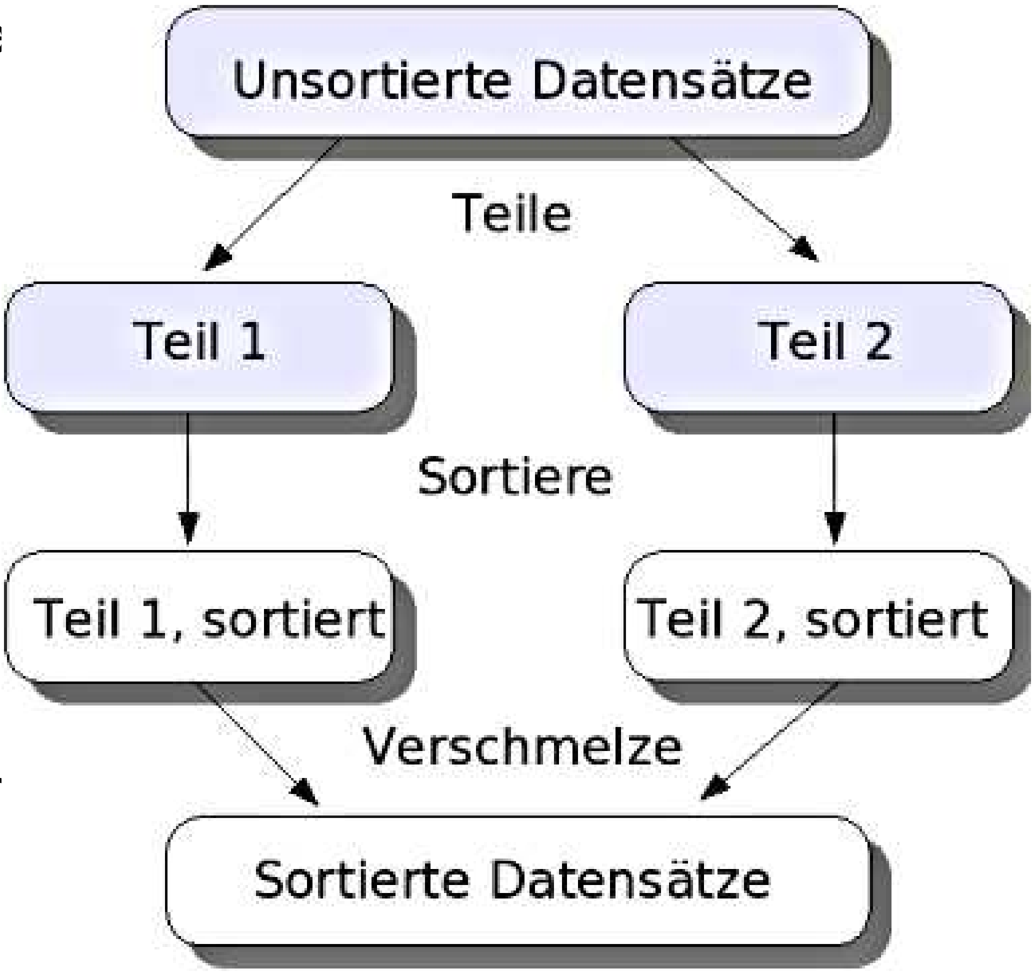
Freitag, 08. Juli 2005

Was es heute zu hören und sehen gibt

- Vom Programm zur Rekurrenz
  - am Beispiel Merge-Sort
- Lösen von Rekurrenzen
  - „von Hand“
  - mit der Mastermethode
  - mit generierenden Funktionen

Merge-Sort

- Gegeben: zu sortierende Datensätze, z.B. Array
- Vorgehen:
  - Teile Daten auf (ggf. mehrfach)
  - sortiere diese
  - verschmelze einzelne Ergebnisse zum Gesamtergebnis
- Besonderheit: beim Aufteilen in Teillisten der Länge 1 entfällt der Zwischenschritt Sortieren



Merge-Sort (II)

8 2 7 1 9 3 6 5

8 2 7 1 | 9 3 6 5

8 2 | 7 1 | 9 3 | 6 5

8 | 2 | 7 | 1 | 9 | 3 | 6 | 5

2 8 | 1 7 | 3 9 | 5 6

1 2 7 8 | 3 5 6 9

1 2 3 5 6 7 8 9

Merge-Sort (III)

```
1. Basisfall.
   if (F == 0 || Length(F)=1) return F;
```

O(1)

```
2. Divide
   // Spalte F in zwei gleich große Listen.
   F1 = CopyList(F,1,Length(F)/2;
   F2 = CopyList(F,(Length(F)/2)+1), Length(F));
```

O(n)

```
3. Conquer
   // Sortiere F1 und F2.
   S1 = MergeSort(F1);
   S2 = MergeSort(F2);
```

???

Merge-Sort (IV)

O(n)

```
4. Join
  // Erzeuge S durch Mischen von S1 und S2.
  while (S1 !== 0 && S2 !== 0)
    if (Head(S1) <= Head(S2))
      Append(S, Head(S1));
      S1 = Remove(Head(S1), S1);
    else
      Append(S, Head(S2));
      S2 = Remove(Head(S2), S2);
  if (S1 == 0)
    AppendList(S, S2);
  else
    AppendList(S, S1);
  return S;
```

Aufwandsanalyse

- Gesamter Aufwand also:
  $O(1) + O(n) + ??? + O(n)$

- Als Rekurrenz:
  $T(n) = 1 + n + T(n/2) + T(n/2) + n$
  $T(n) = 2\,T(n/2) + 2n + 1$

- außerdem: $T(1) = 1$

- Aus geschlossener Form lässt sich der Aufwand ablesen

- hier: $T(n) = O(n \log n)$

Rekurrenzen lösen

1. scharfes Anschauen der Rekurrenz, “finden” einer Lösung und beweisen, dass sie richtig ist

2. Rekurrenz genügt bestimmten Anforderungen → Master-Methode zum lösen verwenden

3. Nichts hilft: Versuch mit generierenden Funktionen

Möglichkeit 1a

- Gegeben: T(0) = 0, T(n) = T(n-1) + n-1
- Idee: Die geschlossene Form muß ein Polynom vom Grad 2 sein, weil:

n	T(n)	Diff1	Diff2
0	0		
1	0	0	
2	1	1	1
3	3	2	1
4	6	3	1

Möglichkeit 1a

- Gegeben: $T(0) = 0$, $T(n) = T(n-1) + n-1$
- Also Ansatz: $f(n) = an^2 + bn + c$
- Einsetzen der ersten Glieder ergibt: $f(n) = {}^1/_2 n^2 - {}^1/_2 n$
- Somit: $T(n) = O(n^2)$

Möglichkeit 1b

- Gegeben: $T(0) = 0$, $T(n) = T(n-1) + n-1$
- Andere Idee: In jedem Schritt wird (n-1) addiert
- Also: $f(n) = 0 + \sum_{i=1}^{n-1} i = ((n-1)n) / 2 = {}^1/_2 n^2 - {}^1/_2 n$
- Und wieder: $T(n) = O(n^2)$

Rekurrenzen lösen

1. scharfes Anschauen der Rekurrenz, "finden" einer Lösung und beweisen, dass sie richtig ist

2. Rekurrenz genügt bestimmten Anforderungen → Master-Methode zum lösen verwenden

3. Nichts hilft: Versuch mit generierenden Funktionen

Möglichkeit 2

- Rekurrenz muß in folgender Form vorliegen:
  $T(n) = a \cdot T(n/b) + f(n)$

Sei $a \geq 1$ und $b \geq 1$, sei $f(n)$ eine Funktion und $T(n)$ gegeben durch die Rekurrenz

$$T(n) = a \cdot T(n/b) + f(n)$$

Dann ist $T(n)$ asymptotisch beschränkt durch:

Wenn $f(n)=O(n^{\log_b a-\varepsilon})$ für einige $\varepsilon>0$, dann $T(n) = \Theta(n^{\log_b a})$

Wenn $f(n)=\Theta(n^{\log_b a})$, dann $T(n) = \Theta(n^{\log_b a} \cdot \log_2 n)$

Wenn $f(n)=\Omega(n^{\log_b a+\varepsilon})$ für einige $\varepsilon>0$ und wenn

$a \cdot f(n/b) \leq c \cdot f(n)$ für $c < 1$ und genügend große n,

dann $T(n) = \Theta(f(n))$.

Möglichkeit 2

- Rekurrenz muß in folgender Form vorliegen:
  $T(n) = a \cdot T(n/b) + f(n)$

- Beispiel:
- $T(n) = 2 \cdot T(n/2) + 2n + 1$
  also: $a = 2$, $b = 2$, $f(n) = 2n+1$

- $\log_2 2 = 1$, also trifft Fall 2 zu, denn $f(n) = 2n+1 = O(n)$
- Nach MT gilt dann: $T(n) = P(n \log_2 n)$

- Master-Theorem liefert keine exakte Lösung für Rekurrenzen!

Rekurrenzen lösen

1. scharfes Anschauen der Rekurrenz, “finden” einer Lösung und beweisen, dass sie richtig ist

2. Rekurrenz genügt bestimmten Anforderungen → Master-Methode zum lösen verwenden

3. Nichts hilft: Versuch mit generierenden Funktionen

Möglichkeit 3

- Gegebene Rekurrenz:
  - $g_0 = 0$
  - $g_1 = 1$
  - $g_n = g_{n-1} + g_{n-2}$

- Zu einer Gleichung zusammenfassen:
  - $g_n = g_{n-1} + g_{n-2} + [n=1]$
  - [query] ist eine spezielle Funktion,
    die 1 liefert, wenn query wahr ist, sonst 0.
  - ausserdem gilt: $g_{n\ (n<0)} = 0$

Rekurrenzen mit generierenden Funktionen lösen

- Mit z^n multiplizieren
  - $g_n z^n = g_{n-1} z^n + g_{n-2} z^n + [n=1] z^n$
- Summieren über n
  - $\sum_n g_n z^n = \sum_n g_{n-1} z^n + \sum_n g_{n-2} z^n + \sum_n [n=1] z^n$
- Vereinfachen
  - $G(z) \quad = \sum_n g_{n-1} z^n + \sum_n g_{n-2} z^n + 0+z+0+0+\ldots$
- Indextransformation durchführen
  - $G(z) \quad = \sum_n g_n z^{n+1} + \sum_n g_n z^{n+2} + z$
  - $G(z) \quad = z\sum_n g_n z^n + z^2\sum_n g_n z^n + z$
  - $G(z) \quad = z\, G(z) + z^2\, G(z) + z$

Rekurrenzen mit generierenden Funktionen lösen

- Nach G(z) auflösen
  - $G(z) \quad = z\sum_n g_n z^n + z^2\sum_n g_n z^n + z$
  - $G(z) \quad = zG(z) + z^2G(z) + z$
  - $G(z) - zG(z) - z^2G(z) = z$
  - $G(z)\,(1 - z - z^2) = z$
  - $G(z) = z / (1 - z - z^2)$

- Koeffizienten von G(z) ermitteln
  - $G(z) = z / (1 - z - z^2)$
  - $G(z) = A / (1 - pz) + B / (1 - qz)$

  - $p = (1 + \sqrt{5}) / 2$ und $q = (1 - \sqrt{5}) / 2$
  - $A = B = 1 / \sqrt{5}$

Rekurrenzen mit generierenden Funktionen lösen

- Folgende Regel gilt:

$$\frac{a}{(1-pz)^{m+1}} = \sum_{n \geq 0} \binom{m+n}{m} ap^n z^n$$

- Vereinfacht gilt also: $a / (1 - pz) = \sum_n ap^n z^n$

- Angewandt auf unser Zwischenergebnis
  - $G(z) = (1 / \sqrt{5}) / (1 – pz) + (1 / \sqrt{5}) / (1 – qz)$
  - mit $p = (1 + \sqrt{5}) / 2$ und $q = (1 - \sqrt{5}) / 2$
- ergibt sich
  - $G(z) = \sum_n (1 / \sqrt{5})\, p^n z^n + \sum_n (1 / \sqrt{5})\, q^n z^n$

Rekurrenzen mit generierenden Funktionen lösen

- $G(z) = \sum_n (1 / \sqrt{5})\, p^n z^n + \sum_n (1 / \sqrt{5})\, q^n z^n$
- Die Koeffizienten des n-ten Summanden lauten jeweils: $(1 / \sqrt{5})\, p^n$ und $(1 / \sqrt{5})\, q^n$
- Die gesuchte geschlossene Form entspricht genau diesen Koeffizienten, d.h.

  $g(n) = (1 / \sqrt{5})\, (p^n + q^n)$

  mit $p = (1 + \sqrt{5}) / 2$ und $q = (1 - \sqrt{5}) / 2$

Und das war's für heute…

Schönes Wochenende & Viel Erfolg bei der (Probe)-Klausur

Folien unter:
http://joachim-wilke.de

Email:
info2@joachim-wilke.de