114

26.01.2012

```
MERGE(A, p, q, r)
n₁ ← q − p + 1
n₂ ← r − q
create arrays L[1 .. n₁ + 1] and R[1 .. n₂ + 1]
for i ← 1 to n₁
      do L[i] ← A[p + i − 1]
for j ← 1 to n₂
      do R[j] ← A[q + j]
L[n₁ + 1] ← ∞
R[n₂ + 1] ← ∞
i ← 1
j ← 1
for k ← p to r
      do if L[i] ≤ R[j]
            then A[k] ← L[i]
                 i ← i + 1
            else A[k] ← R[j]
                 j ← j + 1
```

Wie sieht die Laufzeit aus?

Satz 5.2:

Mergesort hat eine Laufzeit von $O(n \log n)$
(für einen n-elementigen Array A).

Beweis:

Zunächst runden wir n auf die nächste 2er-Potenz.
(in O-Notation ändert sich dadurch nichts, $2^{k-1} < n \leq 2^k$, Faktor 2)
wir erhalten in jedem Schritt also einen Subarray der
Größe $\frac{n}{2}$.
T(n) bezeichne die Laufzeit von Mergesort für einen
n-elementigen Array A.

Divide: $O(1)$

conquer: $2 \cdot T(\frac{n}{2})$

Combine: $O(n)$

Damit erhalten wir folgende Beziehung:

$$T(n) = \begin{cases} O(1) & , n = 1 \\ O(1) + 2 \cdot T(\frac{n}{2}) + O(n) = 2 \cdot T(\frac{n}{2}) + O(n) & , \text{für } n \geq 2 \end{cases}$$

T(n) ist über eine sog. „Rekursionsgleichung" definiert (mehr im nächsten Kapitel).

Für geeignete Konstanten d, g können wir T(n) auch folgendermaßen schreiben.

$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \underbrace{d \cdot n}_{\text{statt } O(n)}, \quad n \geq 2, \quad d \in \mathbb{R}$$

$$T(1) = O(1) = g, \quad g \in \mathbb{R}$$

Nun müssen wir zeigen, dass $T(n) \leq c \cdot n \log n$ für ein geeignetes c.

Induktions-Anfang:

$n = 2$: $T(2) = 2 \cdot T(1) + d \cdot 2 = 2 \cdot g + 2d = 2(g + d)$

$$\underbrace{c}_{2} \cdot \underbrace{n \cdot \log n}_{=1} = 2 \cdot c \geq 2 \cdot (g + d)$$ 117

wähle $c \geq g + d$. (Beachte, dieses impliziert $c \geq d$)

Induktionsvoraussetzung:

Die Behauptung gilt für $k = \frac{n}{2}$, d.h.

$$T\left(\frac{n}{2}\right) \leq c \cdot \frac{n}{2} \cdot \log \frac{n}{2}$$

Induktionsschritt: $\frac{n}{2} \rightsquigarrow n$

$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + d \cdot n$$

$$\leq 2 \left(c \cdot \frac{n}{2} \cdot \log\left(\frac{n}{2}\right)\right) + d \cdot n$$

$$= c \cdot n \cdot \log \frac{n}{2} + d \cdot n$$

$$= c \cdot n \log n \underbrace{- cn + d \cdot n}_{\leq 0,\ \text{da } c \geq d}$$

$$\leq c \cdot n \log n$$

$$\Rightarrow \quad T(n) \in O(n \log n)$$

5.3.3 Das Master-Theorem

Zurück zu

$$T(n) = \sum_{i=1}^{m} T(\alpha_i \cdot n) + \Theta(n^k)$$

Aufteilung in Teilprobleme

Kosten der Aufteilung + Zusammenfügung

Wäre gut, allgemeine Lösung zu kennen...

Satz 5.6 (Master-Theorem)

Sei $T: \mathbb{N} \to \mathbb{N}$ mit

$$T(n) = \sum_{i=1}^{m} T(\alpha_i \cdot n) + \Theta(n^k),$$

wobei $\alpha_i \in \mathbb{R}: 0 < \alpha_i < 1$, $m \in \mathbb{N}$, $k \in \mathbb{N}$.

Dann gilt

$$T(n) \in \begin{cases} \Theta(n^k) & \text{für } \sum_{i=1}^{m} \alpha_i^k < 1 \\ \Theta(n^k \log n) & \text{für } \sum_{i=1}^{m} \alpha_i^k = 1 \\ \Theta(n^c) \text{ mit } \sum_{i=1}^{m} \alpha_i^c = 1 & \text{für } \sum_{i=1}^{m} \alpha_i^k > 1 \end{cases}$$

Beweis: Nicht hier (siehe z.B. Cormen, 4.4)

Beispiele:

(a) $U(n) = 8\,U\left(\frac{n}{3}\right) + n^2$

(Ausprobieren mit $U(1) = 1$:

$U(3) = 8 + 1 = 9$

$U(9) = 8 \cdot 9 + 81 = 17 \cdot 9 = 153$

$U(27) = 8 \cdot 153 + 27^2 < 2 \cdot 27^2$)

Im Master-Theorem:

$\alpha_1 = \ldots = \alpha_8 = \frac{1}{3}$, $m = 8$, $k = 2$

$\sum_{i=1}^{8} \left(\frac{1}{3}\right)^2 < 1$,

also erster Fall:

$U(n) = \Theta(n^2)$

(b) $V(n) = 9 \cdot V\left(\frac{n}{3}\right) + n^2$

Ausprobieren mit $V(1) = 1$:

$$V(1) = 1 = 1 \cdot 3^0$$

$$V(3) = 9 \cdot 1 + 3^2 = 2 \cdot 3^2 = 2 \cdot 3^2$$

$$V(9) = 9 \cdot (2 \cdot 3^2) + 9^2 = 3 \cdot 3^4 = 3 \cdot 9^2$$

$$V(27) = 9 \cdot (3 \cdot 3^4) + 27^2 = 4 \cdot 3^6 = 4 \cdot 27^2$$

usw.!

Im Master-Theorem:

$$\alpha_1 = \dots \alpha_9 = \frac{1}{3}, \quad m = 9, \quad k = 2$$

$$\sum_{i=1}^{9} \left(\frac{1}{3}\right)^2 = 1,$$

also zweiter Fall:

$$V(n) = \Theta\left(n^2 \log n\right)$$

(c) $W(n) = 10 \cdot W(\frac{n}{3}) + n^2$

Ausprobieren:

$W(1) = 1$

$W(3) = 10 + 9 = 19$

$W(9) = 190 + 81 = 271$

$W(27) = 2710 + 729 = 3439$

Wächst noch schneller!

Im Master-Theorem:

$\alpha_1 = \ldots = \alpha_{10} = \frac{1}{3}$, $m = 10$, $k = 2$

$\sum_{i=1}^{10} \left(\frac{1}{3}\right)^2 > 1$

Gesucht also c mit $\sum_{i=1}^{10} \alpha_i^c = 1$,

d.h. $10 \cdot \left(\frac{1}{3}\right)^c = 1$,

also $\left(\frac{1}{3}\right)^c = \frac{1}{10}$, oder $c = \log_3 10 \approx 2{,}096$

Liefert

$W(n) = \Theta\left(n^{2{,}096\ldots}\right)$!

zurück zu Mergesort:

$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + O(n) \qquad \text{für } n \geq 2$$

Parameter des Mastertheorems:

$\alpha_1 = \alpha_2 = \frac{1}{2}$

$m = 2,\ k = 1$

$$\sum_{i=1}^{m} \alpha_i^{k} = 2 \cdot \left(\frac{1}{2}\right)^1 = 1$$

$$\Rightarrow T(n) \in \Theta(n^k \log n) = \Theta(n \log n)$$

$$U(n) = 2\,U\left(\frac{n}{2}\right) + 2n^3 + 4 \cdot U(n/3)$$

$\alpha_1 = \alpha_2 = \frac{1}{2},\ \alpha_3 = \ldots = \alpha_6 = \frac{1}{3}$

$m = 6,\ k = 3$

$$\sum_{i=1}^{m} \alpha_i^{k} = 2\left(\frac{1}{2}\right)^3 + 4 \cdot \left(\frac{1}{3}\right)^3 = 2 \cdot \frac{1}{8} + 4 \cdot \frac{1}{27} = \frac{27+16}{108} = \frac{43}{108} < 1$$

$$\Rightarrow U(n) \in \Theta(n^3)$$