Technische Universität Braunschweig

Institute of Operating Systems
and Computer Networks

Algorithmen und Datenstrukturen
Große Übung #6

Phillip Keldenich, Arne Schmidt

26.02.2017

Heute: Master-Theorem

Vorbetrachtungen

Wir betrachten rekursive Gleichungen der Form

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

D.h. wir haben ein Problem, welches wir in m Teilprobleme unterteilen. Um diese Teilprobleme zu konstruieren und zum Schluss wieder zusammenzuführen benötigen wir $\Theta(n^k)$ Zeit.

Mergesort

Als Beispiel dient Mergesort:

function $\textsc{Mergesort}(A, p, r)$
  if $p < r$ **then**
    $q \leftarrow \lfloor \frac{p+r}{2} \rfloor$
    $\textsc{Mergesort}(A, p, q)$
    $\textsc{Mergesort}(A, q+1, r)$
    $\textsc{Merge}(A, p, q, r)$
  end if
end function

Wir erhalten die folgende Rekursionsgleichung:

$$T(n) = 2T(\frac{n}{2}) + \Theta(n)$$

Dass $T(n) \in \Theta(n \log n)$ gilt, wurde bereits in der Vorlesung per Induktion gezeigt.
Wir wollen die Laufzeit nun anders beweisen.

Master-Theorem

Satz

Sei $T : \mathbb{N} \to \mathbb{R}$ *eine Funktion mit* $T(n) = \sum_{i=1}^{m} T(\alpha_i \cdot n) + \Theta(n^k)$ *mit* $\alpha_i \in \mathbb{R}$, $0 < \alpha_i < 1$, $m \in \mathbb{N}$ *und* $k \in \mathbb{R}$.
Dann ist:

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

Mergesort

Konkret bei Mergesort:

$$T(n) = 2T(\frac{n}{2}) + \Theta(n)$$

Wir haben also $m = 2$, $k = 1$ und $\alpha_1 = \alpha_2 = \frac{1}{2}$
Da $\alpha_1^1 + \alpha_2^2 = 2 \cdot \frac{1}{2} = 1$, folgt nach dem Master-Theorem (Fall 2), dass $T(n) \in \Theta(n \log n)$

Weitere Beispiele (1)

$$U(n) = 13 \cdot U(\frac{n}{4}) + 27n^3 + 27 \cdot U(\frac{n}{16}) + 13n$$

Wir sehen: $m = 40$, $k = 3$, $\alpha_1 = \cdots = \alpha_{13} = \frac{1}{4}$ und $\alpha_{14} = \cdots = \alpha_{40} = \frac{1}{16}$

Wir rechnen

$$\sum_{i=1}^{40} \alpha_i^3 = 13 \cdot \left(\frac{1}{4}\right)^3 + 27 \cdot \left(\frac{1}{16}\right)^3 = \frac{13}{64} + \frac{27}{256 * 16} < \frac{1}{2} + \frac{1}{2} = 1$$

Also ist $U(n) \in \Theta(n^3)$ (Fall 1 des Master-Theorems)

Weitere Beispiele (2)

$$V(n) = 8 \cdot V(\frac{n}{4}) + 18 \cdot V(\frac{n}{6}) + 7n^2 + n \log n + 5$$

Wir sehen: $m = 26$, $k = 2$, $\alpha_1 = \cdots = \alpha_8 = \frac{1}{4}$ und $\alpha_9 = \cdots = \alpha_{26} = \frac{1}{6}$
Wir rechnen

$$\sum_{i=1}^{26} \alpha_i^2 = 8 \cdot \left(\frac{1}{4}\right)^2 + 18 \cdot \left(\frac{1}{6}\right)^2 = \frac{8}{16} + \frac{18}{36} = 1$$

Also ist $V(n) \in \Theta(n^2 \log n)$ (Fall 2 des Master-Theorems)

Weitere Beispiele (3)

$$W(n) = 9 \cdot W(\frac{n}{3}) + 8n + 486 \cdot W(\frac{n}{9}) + 21$$

Wir sehen: $m = 495$, $k = 1$, $\alpha_1 = \cdots = \alpha_9 = \frac{1}{3}$ und

$\alpha_{10} = \cdots = \alpha_{495} = \frac{1}{9}$. Wir rechnen $\sum_{i=1}^{495} \alpha_i = 9 \cdot \frac{1}{3} + 486 \cdot \frac{1}{6} > 1$

3. Fall des Master-Theorems. Wir suchen also nach dem c:

$$\sum_{i=1}^{495} \alpha_i^c = 1 \Leftrightarrow 9 \cdot \frac{1}{3^c} + 486 \cdot \frac{1}{6^c} = 1 \Leftrightarrow \frac{9}{3^c} \cdot (1 + 54 \cdot \frac{1}{3^c}) = 1$$

$$\Leftrightarrow 1 + 54 \cdot \frac{1}{3^c} = \frac{3^c}{9} \Leftrightarrow c = 3$$

Also ist $W(n) \in \Theta(n^3)$

Weitere Beispiele (4)

$$S(n) = 16 \cdot S(\frac{n}{2}) + n^2 + 15n \log n$$

Wir sehen: $m = 16$, $k = 2$, $\alpha_1 = \cdots = \alpha_{16} = \frac{1}{2}$.

Wir rechnen $\sum_{i=1}^{16} \alpha_i^2 = 16 \cdot \left(\frac{1}{2}\right)^2 > 1$

3. Fall des Master-Theorems. Wir suchen also nach dem c:

$$\sum_{i=1}^{16} \alpha_i^c = 1 \Leftrightarrow 16 \cdot \frac{1}{2^c} = 1 \Leftrightarrow 2^c = 16 \Leftrightarrow c = 4$$

Also ist $S(n) \in \Theta(n^4)$

Quicksort - Worst Case

Betrachten wir nun Quicksort:

function $\textsc{Quicksort}(A, p, r)$
  if $p < r$ **then**
    $q \leftarrow \textsc{Partition}(A, p, r)$
    $\textsc{Quicksort}(A, p, q-1)$
    $\textsc{Quicksort}(A, q+1, r)$
  end if
end function

Ist das Master-Theorem direkt anwendbar? Nein! Dafür müsste man wissen, in welchem Verhältnis das Array geteilt wird. Im schlimmsten Fall wäre:

$$T(n) = T(n-1) + \Theta(n)$$

Quicksort - Best Case

Betrachten wir nun Quicksort:

function $\textsc{Quicksort}(A, p, r)$
    if $p < r$ **then**
        $q \leftarrow \textsc{Partition}(A, p, r)$
        $\textsc{Quicksort}(A, p, q-1)$
        $\textsc{Quicksort}(A, q+1, r)$
    end if
end function

Im besten Fall wäre:

$$T(n) = 2 \cdot T(\frac{n}{2}) + \Theta(n)$$

Wie bei Mergesort liefert das Master-Theorem die Laufzeit $\Theta(n \log n)$

Quicksort - Partition

Betrachten wir Quicksort genauer:

function $\textsc{Partition}(A, p, r)$
  $x \leftarrow A[r]$
  $i \leftarrow p - 1$
  for $j \leftarrow p$ **to** $r - 1$ **do**
    if $A[j] \leq x$ **then**
      $i \leftarrow i + 1$
      vertausche $A[i]$ und $A[j]$
    end if
  end for
  vertausche $A[i+1]$ und $A[r]$
  return $i + 1$
end function

Quicksort - Partition

Ein Beispiel: $A = [5, 7, 4, 9, 1, 3, 8, 2, 6]$

Aufruf: $\textsc{Partition}(A,\ 1,\ 9)$:

$x = 6,\ i = 0$. Start der for-Schleife:

$j = 1$: $5 \leq 6 \Rightarrow i = 1$, tausche $A[1]$ und $A[1]$
$j = 2$: $7 \not\leq 6$
$j = 3$: $4 \leq 6 \Rightarrow i = 2$, tausche $A[2]$ und $A[3]$
$j = 4$: $9 \not\leq 6$
$j = 5$: $1 \leq 6 \Rightarrow i = 3$, tausche $A[3]$ und $A[5]$
$j = 6$: $3 \leq 6 \Rightarrow i = 4$, tausche $A[4]$ und $A[6]$
$j = 7$: $8 \not\leq 6$
$j = 8$: $2 \leq 6 \Rightarrow i = 5$, tausche $A[5]$ und $A[8]$

Wir erhalten:

$$A = [5, 4, 1, 3, 2, 6, 8, 7, 9]$$