Datenstrukturen und Algorithmen, WS2024, Übungsblatt 2

Konstantin Krasser

November 3, 2024

Abzugeben bis siehe TC.

Rekursive Laufzeitfunktionen

Hausaufgaben

1. **Aufgabe 1 (4 Punkte).** Berechnen Sie eine scharfe asymptotische obere Schranke für $T(n) = 7T(n/2) + n^2$. Die Basis ist $n < 2$: $T(n) = 1$.

   Lösung:

   Wir verwenden das Master-Theorem, um die Rekurrenz zu lösen.

   Die Rekurrenz hat die Form:

   $$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

   mit $a = 7$, $b = 2$, und $f(n) = n^2$.

   Wir berechnen $\log_b a$:

   $$\log_b a = \log_2 7 \approx 2.807$$

   Wir vergleichen $f(n)$ mit $n^{\log_b a}$:

   $$f(n) = n^2$$
   $$n^{\log_b a} = n^{\log_2 7} \approx n^{2.807}$$

   Da $f(n) = O\left(n^{\log_b a - \epsilon}\right)$ für ein $\epsilon > 0$ (hier $\epsilon \approx 0.807$), befinden wir uns in Fall 1 des Master-Theorems.

   Daher gilt:

   $$T(n) = \Theta\left(n^{\log_b a}\right) = \Theta\left(n^{\log_2 7}\right)$$

   Also ist die scharfe asymptotische obere Schranke:

   $$T(n) = O\left(n^{\log_2 7}\right)$$

2. **Aufgabe 2 (4 Punkte).** Gegeben sei $T(n) = 1$ für $n \leq 2$. Für $n > 2$ gilt: $T(n) = \sqrt{n}\,T(\sqrt{n}) + n$. Zeigen Sie, dass $T(n) \in \mathcal{O}(n \log\log n)$.

   Hinweis: Betrachten Sie zunächst $T'(n) = \frac{T(n)}{n}$, woraus folgt, dass $T'(\sqrt{n}) = \frac{T(\sqrt{n})}{\sqrt{n}}$.

   Lösung:

   Wir definieren $T'(n) = \frac{T(n)}{n}$. Dann gilt $T(n) = n\,T'(n)$.

   Die gegebene Rekurrenz wird zu:

   $$T(n) = \sqrt{n}\,T(\sqrt{n}) + n$$

Dividieren wir beide Seiten durch n:

$$\frac{T(n)}{n} = \frac{\sqrt{n}\,T(\sqrt{n})}{n} + 1$$

$$T'(n) = \frac{T(\sqrt{n})}{\sqrt{n}} + 1 = T'(\sqrt{n}) + 1$$

Wir haben also die Rekurrenz:

$$T'(n) = T'(\sqrt{n}) + 1$$

Um diese zu lösen, betrachten wir die Iteration:

$$\begin{aligned} T'(n) &= T'(\sqrt{n}) + 1 \\ &= T'(\sqrt[4]{n}) + 1 + 1 \\ &= T'(\sqrt[8]{n}) + 1 + 1 + 1 \\ &\vdots \\ &= T'(2) + k \end{aligned}$$

wobei k die Anzahl der Iterationsschritte ist.

Wir stoppen die Iteration, wenn $\sqrt[2^k]{n} \leq 2$, also wenn:

$$\log_2 n^{1/2^k} \leq 1 \implies \frac{\log_2 n}{2^k} \leq 1$$

Daraus folgt:

$$2^k \geq \log_2 n \implies k \geq \log_2 \log_2 n$$

Damit ist:

$$T'(n) = T'(2) + k = T'(2) + \log_2 \log_2 n$$

Da $T'(2)$ eine Konstante ist, gilt:

$$T'(n) = \mathcal{O}(\log \log n)$$

Da $T(n) = n\,T'(n)$, folgt:

$$T(n) = n\,\mathcal{O}(\log \log n) = \mathcal{O}(n \log \log n)$$

3. **Aufgabe 3 (4 Punkte).** Berechnen Sie eine scharfe asymptotische obere Schranke für die folgende Rekurrenzfunktion $T(n) = T(\sqrt{n}) + \log_2 \log_2 n$. Nehmen Sie an, dass $T(n) = 1$ für $n \leq 2$.

   Hinweis: Verwenden Sie die Substitution $m = \log_2 n$, ähnlich wie in der Übung.

   Lösung:

   Wir setzen $m = \log_2 n$. Dann ist $n = 2^m$ und $\sqrt{n} = 2^{m/2}$. Außerdem gilt $\log_2 \log_2 n = \log_2 m$.

   Die Rekurrenz wird zu:

   $$T(2^m) = T\left(2^{\frac{m}{2}}\right) + \log_2 m$$

   Definieren wir $S(m) = T(2^m)$, erhalten wir:

   $$S(m) = S\left(\frac{m}{2}\right) + \log_2 m$$

Wir lösen diese Rekurrenz durch Iteration:

$$
\begin{aligned}
S(m) &= S\left(\frac{m}{2}\right) + \log_2 m \\
&= S\left(\frac{m}{4}\right) + \log_2\left(\frac{m}{2}\right) + \log_2 m \\
&= S\left(\frac{m}{8}\right) + \log_2\left(\frac{m}{4}\right) + \log_2\left(\frac{m}{2}\right) + \log_2 m \\
&\vdots \\
&= S\left(\frac{m}{2^k}\right) + \sum_{i=0}^{k-1} \log_2\left(\frac{m}{2^i}\right)
\end{aligned}
$$

Wir stoppen die Iteration, wenn $\frac{m}{2^k} \leq 1$, also wenn $k \geq \log_2 m$.
Die Summe wird dann:

$$\sum_{i=0}^{k-1} (\log_2 m - i) = k \log_2 m - \frac{k(k-1)}{2}$$

Da $k = \lceil \log_2 m \rceil$, ist $k = \mathcal{O}(\log m)$.
Also ist:

$$S(m) = S(1) + \mathcal{O}\left((\log m) \log m - (\log m)^2\right) = \mathcal{O}\left((\log m)^2\right)$$

Da $S(1)$ eine Konstante ist, folgt:

$$T(n) = S(m) = \mathcal{O}\left((\log \log n)^2\right)$$