Datenstrukturen und Algorithmen, WS2024, Übungsblatt 2
Rekursive Laufzeitfunktionen

Konstantin Krasser

October 17, 2024

1. **Aufgabe 1 (4 Punkte).** Sei $T(n) = 1$ für $n \leq 2$. Für $n > 2$ gilt: $T(n) = T(n-2) + \log n$. Zeigen Sie: $T \in \mathcal{O}(n \log n)$. Sie können annehmen, dass n gerade ist.

   Lösung:

   Um zu zeigen, dass $T(n) \in \mathcal{O}(n \log n)$, benutzen wir eine Induktion und eine Abschätzung der Summe:

   (a) **Basisfall:** Für $n \leq 2$ ist $T(n) = 1$, was offensichtlich eine konstante Laufzeit ist und somit kleiner als $\mathcal{O}(n \log n)$ für kleine n.

   (b) **Induktionsschritt:** Für $n > 2$ gilt:

   $$T(n) = T(n-2) + \log n$$

   Um $T(n)$ abzuschätzen, summieren wir die Terme bis zum Basisfall:

   $$\log(2k) = log(2) + log(k)$$

   $$T(n) = 1 + \sum_{k=2}^{n/2} \log(2k)$$

   Diese Summe lässt sich durch $n \log n$ abschätzen. Daher folgt:

   $$T(n) \in \mathcal{O}(n \log n)$$

2. **Aufgabe 2 (4 Punkte).** Berechnen Sie eine scharfe asymptotische obere Schranke für $T(n) = 30T(n/3) + n^3$. Es gilt für den Basisfall $n < 2 : T(n) = 1$.

   Lösung:

   Wir wenden das Master-Theorem an, um eine scharfe obere Schranke zu finden:

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

   mit $a = 30$, $b = 3$, und $f(n) = n^3$.

   (a) Berechnen Sie den Exponenten $\log_b a$:

   $$\log_3 30 \approx 3.096$$

(b) Vergleichen Sie $f(n) = n^3$ mit $n^{\log_3 30}$:
Da $f(n)$ (n^3) kleiner ist als $n^{\log_3 30}$, befinden wir uns im Fall 1 des Master-Theorems:

$$T(n) = \Theta\left(n^{\log_3 30}\right)$$

Also ist die scharfe asymptotische obere Schranke:

$$T(n) = \mathcal{O}\left(n^{3.096}\right)$$

3. **Aufgabe 3 (4 Punkte).** Berechnen Sie eine möglichst knappe obere Schranke der folgenden rekursiven Funktion $T(n)$: $T(n) = 2T(\sqrt{n}) + \log_2 n$, mit $T(n) = \mathcal{O}(1)$ für $n \leq 2$.

Lösung:

Wir wenden eine Veränderliche-Substitution an: Setzen Sie $m = \log_2 n$, dann ist $n = 2^m$. Die Rekursion wird umgeschrieben als:

$$T(2^m) = 2T(2^{m/2}) + m$$

Setzen Sie $S(m) = T(2^m)$. Dann haben wir:

$$S(m) = 2S(m/2) + m$$

Diese Rekursion entspricht der Form des Master-Theorems mit $a = 2$, $b = 2$, und $f(m) = m$. Da $f(m) = m = \Theta(m^1)$, haben wir $d = 1$. Der kritische Exponent ist:

$$\log_b a = \log_2 2 = 1$$

Da $f(m)$ der gleichen Ordnung ist (m^1), befinden wir uns im Fall 2 des Master-Theorems:

$$S(m) = \Theta(m \log m)$$

Daher ist $T(n) = S(\log_2 n) = \Theta(\log n \cdot \log \log n)$. Eine möglichst knappe obere Schranke ist also:

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