Datenstrukturen und Algorithmen: Hausübung 2

Felix Schrader, 3053850
Jens Duffert, 2843110
Eduard Sauter, 3053470

29. Oktober 2015

Aufgabe 1

a) Es gilt:

$$2^{n+d} = 2^d \cdot 2^n$$

Wählt man $c = 42 \cdot 2^d$ und $n_0 = 1$, so ist

$$f(n) = 2^d \cdot 2^n \leq c \cdot g(n) = 42 \cdot 2^d \cdot 2^n$$

für alle $n \geq n_0$, da man auf beiden Seiten durch $2^d \cdot 2^n$ teilen kann und die wahre Aussage $1 \leq 42$ erhält.

b) Es gilt:

$$\log_a(n) = \frac{1}{\log_b(a)} \cdot \log_b(n)$$

In der Stundenübung 2.1 haben wir gezeigt:

$$\mathcal{O}(f(n)) = \mathcal{O}(c \cdot f(n)), \quad c > 0$$

Wählt man also $f(n) = \log_a(n)$ und $c = 1/\log_b(a) > 0$, so erhält man:

$$\mathcal{O}(\log_a(n)) = \mathcal{O}(\log_b(n))$$

c) In der Vorlesung wurde als Rechenregel für Landau-Symbole eingeführt:

$$\Theta(f(n)) + \Theta(g(n)) = \Theta(f(n) + g(n))$$

Indem man diese Regel iterativ anwendet, erhält man für die Summe:

$$\begin{aligned}\sum_{i=1}^{n} \Theta(i) &= \Theta\left(\sum_{i=1}^{n} i\right) \\ &= \Theta\left(\frac{n(n+1)}{2}\right) \\ &= \Theta\left(\frac{n^2}{2} + \frac{n}{2}\right)\end{aligned}$$

Wählt man $c = 1$ und $n_0 = 1$, so erkennt man:

$$\begin{aligned}
&\frac{n^2}{2} + \frac{n}{2} \in \mathcal{O}(n^2), \quad \text{da} \\
&\frac{n^2}{2} + \frac{n}{2} \leq n^2 \quad \forall n \geq n_0
\end{aligned}$$

Mit $c = 1/2$ und $n_0 = 1$ sieht man außerdem:

$$\begin{aligned}
&\frac{n^2}{2} + \frac{n}{2} \in \Omega(n^2), \quad \text{da} \\
&\frac{n^2}{2} + \frac{n}{2} \geq \frac{n^2}{2} \quad \forall n \geq n_0 \\
&\Leftrightarrow \frac{n}{2} \geq 0 \quad \forall n \geq n_0
\end{aligned}$$

Also ist insgesamt:

$$\begin{aligned}
\sum_{i=1}^{n} \Theta(i) &= \Theta\left(\frac{n^2}{2} + \frac{n}{2}\right) \\
&= \Theta(n^2)
\end{aligned}$$

d) Es gilt:

$$\begin{aligned}
\sum_{r=0}^{n-1}\sum_{s=r}^{n-1}\sum_{i=r}^{s} 1 &= \sum_{r=0}^{n-1}\sum_{s=r}^{n-1}(s-r+1) \\
&= \sum_{r=0}^{n-1}\sum_{s=0}^{n-r-1}(1+s) \\
&= \sum_{r=0}^{n-1}(n-r) + \frac{(n-r-1)(n-r)}{2} \\
&= \sum_{r=0}^{n-1}\left(\frac{n^2+n}{2} - \frac{2n+1}{2}r + \frac{r^2}{2}\right) \\
&= \frac{n(n^2+n)}{2} - \frac{n(n-1)(2n+1)}{4} + \frac{n(n-1)(2n-1)}{12} \\
&= \frac{n^3}{6} + \frac{n^2}{2} + \frac{n}{3}
\end{aligned}$$

Wählt man $c = 1/6$ und $n_0 = 1$, so erkennt man:

$$\begin{aligned}
&\sum_{r=0}^{n-1}\sum_{s=r}^{n-1}\sum_{i=r}^{s} 1 \in \Omega(n^3) \\
&\Leftrightarrow \frac{n^3}{6} + \frac{n^2}{2} + \frac{n}{3} \geq \frac{n^3}{6} \quad \forall n \geq n_0 \\
&\Leftrightarrow \frac{n^2}{2} + \frac{n}{3} \geq 0 \quad \forall n \geq n_0
\end{aligned}$$

Aufgabe 2

Gegeben ist:

$$f(n) \in \mathcal{O}(g(n)) \Leftrightarrow \lim_{n\to\infty} \sup \frac{f(n)}{g(n)} < \infty$$

$$f(n) \in \Omega(g(n)) \Leftrightarrow \lim_{n\to\infty} \inf \frac{f(n)}{g(n)} > 0$$

a) Die Funktion die gegeben ist sieht wie folgt aus:

$$e^n \notin \mathcal{O}(n^c) \quad \forall c \in \mathrm{N}$$

Es wird nun der Limes betrachtet.

$$\lim_{n\to\infty} e^n = \lim_{n\to\infty} n^c = \infty$$

Da beide Grenzwerte ∞ betragen, wird L'Hospital verwendet.

$$\lim_{n\to\infty} \frac{e^n}{n^c} = \lim_{n\to\infty} \frac{e^n}{c \cdot n^{c-1}} = \lim_{n\to\infty} \frac{e^n}{c \cdot (c-1) \cdot n^{c-2}} = \ldots$$
$$= \lim_{n\to\infty} \frac{e^n}{c \cdot (c-1) \cdot \ldots \cdot n^{c-c}} = \lim_{n\to\infty} \frac{e^n}{c!}$$

Da der Limes ∞ ist, folgt daraus das die Funktion $e^n \notin \mathcal{O}(n^c)$ ist (nach der oben genannten Definition.)

b) Die Funktion die gegeben ist sieht wie folgt aus:

$$\log(n) \notin \Omega(n)$$

Wie im Hinweis geschrieben, werden die Limites von Zähler und Nenner betrachtet.

$$\lim_{n\to\infty} \log(n) = \lim_{n\to\infty} n = \infty$$

Nun wird L'Hospital angewendet.

$$\lim_{n\to\infty} \frac{\log(n)}{n} = \lim_{n\to\infty} \frac{\frac{1}{n}}{1} = \lim_{n\to\infty} \frac{1}{n}$$

Es wird der Limes nun berechnet:

$$\lim_{n\to\infty} \frac{1}{n} = 0$$

Da der Limes 0 ist folgt daraus, dass die Funktion $\log(n) \notin \Omega(n)$ ist (nach der oben genannten Definition).

Aufgabe 3

Es sei $f(n)$ die Laufzeitfunktion des jeweiligen Algorithmus.

`Linear-Search`

Best-Case Das gesuchte Element ist an erster Position. Die Laufzeit ist folglich $\Theta(1)$, da nur ein Element geprüft wird.

Worst-Case Das gesuchte Element ist an letzter Position. Es werden also n Elemente überprüft. Dies bedeutet $\Theta(n)$.

Average-Case Für einen beliebigen Index i ist die Wahrscheinlichkeit, den Wert x an dieser Stelle zu finden nach Vorraussetzung $\frac{1}{n}$. Es sei j der Index von x. Also $P(j = i) = \frac{1}{n}$. Damit ist der Erwartungswert

$$E(j) = \sum_{i=0}^{n-1} (i+1) P(j = i) = \frac{1}{n} \sum_{i=1}^{n} i = \frac{n+1}{2}$$

Also ist die Laufzeitfunktion von `Linear-Search` in $\Theta(n)$.

`Randomized-Search`

Best-Case Der erste Index ist nach dem ersten Versuch derjenige von x. Also ist in diesem Fall wieder $f(n) \in \Theta(1)$.

Worst-Case Der Wert wird nie gefunden, weil zum Beispiel der Zufallsgenerator in jedem Iterationsschritt 1 ausgibt und x nicht den ersten Index belegt.

Average-Case Es sei q die Anzahl der benötigten Schritte um x zu finden. In jedem Schritt ist die Wahrscheinlichkeit x zu finden nach Vorraussetzung wieder $\frac{1}{n}$. Für $P(q = i)$ muss noch die Wahrscheinlichkeit berücksichtigt werden, dass die Iteration nicht abgebrochen ist, weil x bereits gefunden wurde. Diese ist nach i Schritten $\left(1 - \frac{1}{n}\right)^i$. Da die ersten $i - 1$ Zufallsexperimente unabhängig von den i-ten Versuch sind, ist

$$P(q = i) = \frac{1}{n} \left(1 - \frac{1}{n}\right)^{i-1}$$

Damit ergibt sich der Erwartungswert zu

$$\begin{aligned} E(q) &= \sum_{i=1}^{\infty} i P(q = i) \\ &= \left(1 - \frac{1}{n}\right)^{-1} \sum_{i=1}^{\infty} \frac{i}{n} \left(1 - \frac{1}{n}\right)^i \\ &= \frac{1}{1 - \frac{1}{n}} \frac{1 - \frac{1}{n}}{\left(\frac{1}{n}\right)^2} = n^2 \end{aligned}$$

Damit ist $f(n) \in \Theta(n^2)$