Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. F. Kuhn
M. Fuchs, G. Schmid

Algorithmen und Datenstrukturen
Sommersemester 2024
Übungsblatt 2

Abgabe: Dienstag, 30. April, 2024, 10:00 Uhr

Aufgabe 1: $\mathcal{O}$-Notation *(9 Punkte)*

Beweisen oder widerlegen Sie die folgenden Aussagen anhand der *Mengendefinition* der $\mathcal{O}$-Notation (Vorlesungsfolien Woche 2, Folie 11 und 12).

(a) $2n^3 - 5 \cdot n^2 + 1 \in \mathcal{O}(n^4)$ nicht schneller als $\xrightarrow{\checkmark}$ $n^4 = n^3$ *(1 Punkt)*

(b) $\log_3(n) \in o(\log_5(n))$ langsamer als $\rightarrow$ log = log *(2 Punkte)*

(c) $n! \in \Omega(2^n)$ mind. so schnell $\xrightarrow{\checkmark}$ $\underset{\text{Faktoriell}}{n!} > \overset{\text{quadratisch}}{2^n}$ *(2 Punkte)*

(d) $\log_2\left(n^2\right) \in \omega((\log_2 n)^2)$ schneller als $\xrightarrow{\times}$ $\log_2(n^2) < \log_2(n)^2$ log < quadratisch *(2 Punkte)*

(e) $\max\{f(n), g(n)\} \in \Theta(f(n) + g(n))$ für nicht negative Funktionen f und g. gleich schnell *(2 Punkte)*

Aufgabe 2: Sortieren nach Asymptotischem Wachstum *(4 Punkte)*

Sortieren Sie folgende Funktionen nach asymptotischem Wachstum. Schreiben Sie $g <_{\mathcal{O}} f$ falls $g \in \mathcal{O}(f)$ *und* $f \notin \mathcal{O}(g)$. Schreiben Sie $g =_{\mathcal{O}} f$ falls $f \in \mathcal{O}(g)$ *und* $g \in \mathcal{O}(f)$ (kein Beweis nötig). nicht schneller

$\sqrt{n} \cdot n^{3/2}$ – quadratisch	3^n – Exponentiell	$\frac{1}{4} \cdot n!$ – Faktoriell	$27 \cdot n$ – Linear
$4^{n/2}$	$100 \cdot n^{100}$	$\log_2(n^3)$	n^n
$12 \cdot \sqrt{\log_2(n)}$	$21 \cdot \log_2(\sqrt{n})$	$(n-1)!$	$\log_2(n^n)$ Logarithmisch

Aufgabe 3: k-MergeSort *(7 Punkte)*

In Übungsblatt 1 ging es darum eine Variante des Mergesort Algorithmus zu implementieren, welche für einen gegebenen Parameter $k > 1$, das gegebene Array in k Teile der Größen $O(n/k)$ zerlegt, wobei n die Größe des Arrays ist. Wir wollen hier nun die Laufzeit dieser Variante analysieren.

a) Sei $T(n, k)$ die Laufzeit für obigen Algorithmus mit Parametern n und k. Geben Sie eine rekursive Formel für $T(n, k)$ an (analog zur Folie 24, Foliensatz 2). Zur Einfachheit können Sie annehmen dass der Algorithmus das Array in jedem rekursiven Schritt in k Teile der Größe genau n/k teilt. *(3 Punkte)*

b) Zeigen Sie mithilfe von vollständiger Induktion dass $T(n, k) = \mathcal{O}(\log_k(n) \cdot n \cdot k)$. *(3 Punkte)*

c) Setzen Sie die Werte $2, 3, \log_2(n), n/4$ für k ein. Für welche dieser Werte ist die Laufzeit asymptotisch am besten? *(1 Punkt)*

Aufgabe 1: $\mathcal{O}$-Notation (9 Punkte)

Beweisen oder widerlegen Sie die folgenden Aussagen anhand der *Mengendefinition* der $\mathcal{O}$-Notation (Vorlesungsfolien Woche 2, Folie 11 und 12).

(a) $2n^3 - 5 \cdot n^2 + 1 \in \mathcal{O}(n^4)$ nicht schneller als $\overset{\checkmark}{\rightarrow}$ $n^4 = n^3$ (1 Punkt)

(b) $\log_3(n) \in o(\log_5(n))$ langsamer als $\rightarrow$ log = log (2 Punkte)

(c) $n! \in \Omega(2^n)$ mind. so schnell $\overset{\checkmark}{\rightarrow}$ $n! > 2^n$ Fakultät, quadratisch (2 Punkte)

(d) $\log_2(n^2) \in \omega((\log_2 n)^2)$ schneller als $\overset{\times}{\rightarrow}$ $\log_2(n^2) < \log_2(n)^2$ log < quadratisch (2 Punkte)

(e) $\max\{f(n), g(n)\} \in \Theta(f(n) + g(n))$ für nicht negative Funktionen f und g. gleich schnell (2 Punkte)

a) $f(n) = 2n^3 - 5n^2 + 1 \quad g(n) = n^4$

Seien $n_0 = 1$, $c = 8$. Für alle $n \geq n_0$: $n^4 \geq n^3 \geq n^2 \geq 1$

Dreiecksungleichung

$$|2n^3 - 5n^2 + 1| \leq |2n^3| + |-5n^3| + |n^3| \leq cn^4$$

$\hookrightarrow f(n) \in O(g(n))$ wahr

b) Seien $n_0 = 1$, $c = 1$ Für alle $n \geq n_0$: $\log_3 n \geq \log_5 n$

$$\underbrace{\log_3}_{f(n)} \nleq \underbrace{\log_5}_{g(n)} \cdot c$$

$\hookrightarrow f(n) \notin o(g(n))$ falsch

c) $n! \in \Omega(2^n)$

Seien $n_0 = 1$ und $c = 1$ Für alle $n \geq n_0$: $n! \geq 2^n$

$$n! \geq 2^n \cdot 1$$

$\hookrightarrow f(n) \in \Omega(g(n))$ wahr

d) Sei $n_0 = 1$, $c > 0$ Für alle $n \geq n_0$ $n^2 \geq n$

$$\log_2 n^2 = \underbrace{2 \cdot \log_2 n}_{f(n)} \overset{!}{\geq} \underbrace{(\log_2 n)^2 \cdot c}_{g(n)} \quad | : \log_2 n$$

$$\Leftrightarrow 2 \ngeq \log_2 n \cdot c$$

$\hookrightarrow f(n) \notin \omega(g(n))$

$\Theta(g(n)) := O(g(n)) \cap \Omega(g(n))$

- Funktion $f(n) \in \Theta(g(n))$, falls es Konstanten c_1, c_2 und n_0 gibt, so dass $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ für alle $n \geq n_0$, resp. falls $f(n) \in O(n)$ und $f(n) \in \Omega(n)$

Sei $n_0 = 1$ $\quad n \geq n_0 \quad f(n), g(n) \geq 0$

$$\underbrace{c_1 \cdot (g(n) + f(n))}_{g_1(n)} \overset{!}{\leq} \underbrace{\max(g(n), f(n))}_{f(n)} \overset{!}{\leq} \underbrace{(g(n) + f(n)) \cdot c_2}_{g_2(n)}$$

wenn $c_1 \leq \frac{1}{2}$, dann ist $g(n) + f(n) \cdot c_1$ quasi kleiner gleich der Durchschnitt von g und f

und $(g(n) + f(n)) \cdot c_1 \leq \max(g(n), f(n))$

wenn $c_2 \geq 1$, dann gilt, dass $(g(n) + f(n)) \cdot c_2 \geq \max(f(n), g(n))$

$\hookrightarrow \quad f(n) \in O(g_2(n))$ und $f(n) \in \Omega(g_1(n)) \Rightarrow f(n) \in \Theta(g(n))$

wahr

$$T(n,k) \leq \sum_{i=0}^{k-1} T\left(\left\lfloor \frac{n}{k} \right\rfloor\right) + T\left(\left\lfloor \frac{n}{k} \right\rfloor + n \% k\right) + b \cdot n$$

$$T(1) \leq b$$

IA

$$T(n,2) \overset{\checkmark}{=} O(\log_2(n) \cdot n \cdot 2)$$

IV

wir setzen voraus $k = m$

$$T(n,m) = O(\log_m(n) \cdot n \cdot m)$$

IS

Angenommen, $k = m+1$

$$T(n,m+1) = O(\log_{m+1}(n) \cdot n(m+1))$$

???

$T(n,2) = O(\log_2(n) \cdot n \cdot 2)$

$T(n,3) = O(\log_3(n) \cdot n \cdot 3)$

$T(n, \log_2(n)) = O(\log_{\log_2(n)}(n) \cdot n \cdot \log_2(n))$

$T(n, \frac{n}{4}) = O(\log_{\frac{n}{4}}(n) \cdot n \cdot \frac{n}{4})$

$k=3$ ist asymptotisch am besten.

Yorick Schiffer - ys151
Paul Tröster - pt144
Ege Tekin - et130