Antwort für Übungsblatt 2

Jian Dong jd81vuti
Zezhi Chen zc75diqa
Hanyu Sun hs54keri

May 2, 2019

1 P1 (Gruppendiskussion)

(a) Asymptotische Notation(O,o,Ω, ω, Θ)
$O(g) = \{f : \exists c \in \mathbb{R}_{>0}, n_0 \in \mathbb{N}, \forall n \geq n_0, 0 \leq f(n) \leq cg(n)\}$
$o(g) = \{f : \forall c \in \mathbb{R}_{>0}, \exists n_0 \in \mathbb{N}, \forall n \geq n_0, 0 \leq f(n) < cg(n)\}$
$O(g)$ bedeutet, dass die Ordung von $g(n)$ gleich oder grosser als $f(n)$ sein muss,aber $o(g)$ bedeutet, dass die Ordnung von $g(n)$ grosser als $f(n)$ sein muss.

$\Omega(g) = \{f : \exists c \in \mathbb{R}_{>0}, n_0 \in \mathbb{N}, \forall n \geq n_0, 0 \leq cg(n) \leq f(n)\}$
$\omega(g) = \{f : \forall c \in \mathbb{R}_{>0}, \exists n_0 \in \mathbb{N}, \forall n \geq n_0, 0 \leq cg(n) < f(n)\}$
$\Omega(g)$ bedeutet, dass die Ordung von $g(n)$ gleich oder kleiner als $f(n)$ sein muss,aber $\omega(g)$ bedeutet, dass die Ordnung von $g(n)$ kleiner als $f(n)$ sein muss.

$\Theta(g) = \{f : \exists c_1, c_1 \in \mathbb{R}_{>0}, n_0 \in \mathbb{N}, \forall n \geq n_0, 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)\}$
$\Theta(g)$ bedeutet, dass $g(n)$ und $f(n)$ in gleiche Ordnung sind.

(b) Divide-and-Conquer Ansatz
Zerlege das Probleme in mehrere Teilprobleme, die loeserbar sind.

(c) Bubblesort,Mergesort,Quicksort
Bubblesort: 1.Vergleiche Paare von benachbarten Schluesselwerten.
2.Tausche das Paare, falls rechter Schluesselwert kleiner ist als linker.

Mergesort: 1.Rekusiv teilt die Folge in n Elementen in zwei Teilfolgen von je n/2 Elementen auf bis jede Teilfolgen nach Reihenolge anordnende sind.
2.Die zwei sortierten Teilfolgen,die am Endstuecke liegen, mischen, um die sortierte Loesung zu erzeugen.

Quicksort: 1.Zerlege den Array A[p..r] in zwei Teilarrays A[p..q-1] und A[q+1..r], sodass jedes Element von A[p..q-1] kleiner oder gleich A[q] ist(die Zahlenwert von A[q] ist die Zahlenwert von A[r] in originale Array), welches wiederum kleiner oder gleich jedem Element von A[q+x..r] ist. Berechnen den Index q als teil vom Partion Algorithmus.
2.Sortieren beider Teilarrays A[p..q-1] und A[q+1..r] durch rekursiven Aufruf von Quicksort.

2 P2 (Rechenregeln für asymptotische Notation)

(a) i) f(n)=O(g(n)) genau dann wenn $g(n) = \Omega(f(n))$
$f(n) = O(g(n)) \Leftrightarrow$ Die Ordnung von f(n) gleich oder kleiner als g(n)$\Leftrightarrow$Die Ordnung von g(n) gleich oder grosser als f(n)$\Leftrightarrow$ $g(n) = \Omega(f(n))$
(a) ii) f(n)=o(g(n)) genau dann wenn $g(n) = \omega(f(n))$
$f(n) = o(g(n)) \Leftrightarrow$ Die Ordnung von f(n) kleiner als g(n)$\Leftrightarrow$Die Ordnung von g(n) grosser als f(n)$\Leftrightarrow$ $g(n) = \omega(f(n))$

(b) $o(g(n)) \subseteq O(g(n))$**,und** $\omega(g(n)) \subseteq \Omega(g(n))$
$o(g(n)) + \Theta(g(n)) = O(g(n)) \Rightarrow o(g(n)) \subseteq O(g(n))$
$\omega(g(n)) + \Theta(g(n)) = \Omega(g(n)) \Rightarrow \omega(g(n)) \subseteq \Omega(g(n))$

(c) $O(g(n)) \cap \Omega(g(n)) = \Theta(g(n))$**, und** $o(g(n)) \cap \Omega(g(n)) = \varnothing$
$O(g(n)) \cap \Omega(g(n))$ bedeutet, dass die beide Funktion in gleich Ordnung sind $\Leftrightarrow \Theta(g(n))$
$o(g(n)) \cap \Omega(g(n))$ bedeutet, waehrend die Ordnung von f(n) kleiner als g(n) ist, muss die Ordnung von f(n) grosser als g(n) ist.$\Leftrightarrow \varnothing$

(d) i) ist$f_1(n) = O(g_1(n))$ **und** $f_2(n) = O(g_2(n))$**, dann gilt** $f_1(n)+f_2(n) = O(mas(g_1(n), g_2(n)))$
Denn die Ordnung von$f_1(n) + f_2(n)$ haengt von die groesste Ordnung von f_1 und f_2 ab. und die groesste Ordnung von f_1 und f_2 gleich oder kleiner als $O(mas(g_1(n) * g_2(n)))$ ist, d.h.$f_1(n) + f_2(n) = O(mas(g_1(n), g_2(n)))$
(d) ii) ist$f_1(n) = O(g_1(n))$ **und** $f_2(n) = O(g_2(n))$**, dann gilt** $f_1(n)*f_2(n) = O(g_1(n) * g_2(n))$
Nehmen wir an,dass die Ordnung von$f_1(n), f_2(n), g_1(n), g_2(n)$ ist $j, k, l, m(j \leq l, k \leq m)$. Die Ordnung von $f_1(n) * f_2(n)$ ist $j * k$. Die Ordnung von $g_1(n) * g_2(n)$ ist $l * m$.Denn $j \leq l, k \leq m$ und $j, k, l, m > 0$, so $j * k \leq l * m$, d.h.$f_1(n) * f_2(n) = O(g_1(n) * g_2(n))$
(d) iii) Es gilt $f(n) = O(f(n))$
$f(n)$ und $f(n)$ ist in gleiche Ordnung,d.h.$\exists c_1, c_1 \in \mathbb{R}_{>0}, n_0 \in \mathbb{N}, \forall n \geq n_0, 0 \leq c_1 f(n) \leq f(n) \leq c_2 f(n)\}$. Dann$f(n) = \Theta(f(n)) \in O(f(n)) \Rightarrow f(n) = O(f(n))$
(d) iv) Ist $f(n) = O(g(n))$ **und** $g(n) = O(h(n))$**, dann gilt** $f(n) = O(h(n))$**.**
$f(n) = O(g(n)) \Leftrightarrow$ die Ordnung von f(n) gleich oder kleiner als g(n).$g(n) = O(h(n)) \Leftrightarrow$ die Ordnung von g(n) gleich oder keliner als h(n),d.h. Die Ordnung von f(n) gleich oder kleiner als h(n)$\Leftrightarrow$ $f(n) = O(h(n))$

3 P3 (Rechnen mit asymptotischer Notation)

f(n)	g(n)	O	o	Ω	ω	Θ
$log^k(n)$	n^ε	x	x			
n^k	c^n	x	x			
2^n	$2^{n/2}$			x	x	
$n^{log(c)}$	$c^{log(n)}$	x		x		x
n^r	n^s	x	x			
$log(n!)$	$log(n^n)$	x	x			

1. $\lim\limits_{n\to\infty} \frac{log_n(log^k(n))}{log_n(n^\varepsilon)} = \frac{klog_n(log(n))}{\varepsilon} = \frac{klog(log(n))}{\varepsilon log(n)} = 0$ d.h. die Ordnung von f(n) kleiner als g(n).
2. $\lim\limits_{n\to\infty} \frac{log(n^k)}{log(c^n)} = \frac{klog(n)}{nlog(c)} = 0$ d.h. die Ordnung von f(n) kleiner als g(n).
3. $\lim\limits_{n\to\infty} \frac{2^n}{2^{n/2}} = 2^{n/2} = +\infty$ d.h. die Ordnung von f(n) grosser als g(n).
4. $\lim\limits_{n\to\infty} \frac{log(n^{log(c)})}{log(c^{log(n)})} = \frac{log(c)log(n)}{log(n)log(c)} = 1$ d.h. im gleichen Ordnung.
5. $\lim\limits_{n\to\infty} \frac{n^r}{n^s} = n^{r-s} = 0$ d.h. die Ordnung von f(n) kleiner als g(n).
6. $\lim\limits_{n\to\infty} (log(n!) - log(n^n)) = log(\frac{n!}{n^n}) < log(1/n) = -\infty$ d.h. die Ordnung von f(n) kleiner als g(n).

4 P4 (Darstellung von Merge Sort)

$$[14, 9, 5, 8, 11, 4, 21, 7, 6]$$
$$[9, 14, 5, 8, 11, 4, 21, 7, 6]$$
$$[5, 9, 14, 8, 11, 4, 21, 7, 6]$$
$$[5, 9, 14, 8, 11, 4, 21, 7, 6]$$
$$[5, 8, 9, 11, 14, 4, 21, 7, 6]$$
$$[5, 8, 9, 11, 14, 4, 21, 7, 6]$$
$$[5, 8, 9, 11, 14, 4, 21, 6, 7]$$
$$[5, 8, 9, 11, 14, 4, 6, 7, 21]$$
$$[4, 5, 6, 7, 8, 9, 11, 14, 21]$$

5 P5 (Bubble Sort)

pass