2. Tutorium - Algorithmen I

Nina Zimbel

29.04.2015

- Übungsblätter
- O-Kalkül
- Merging-Problem

Zu den Übungsblättern

- ab nächstem Übungsblatt Deckblatt nötig
- Webinscribe Deckblattgenerator: https://webinscribe.ira.uka.de/deckblatt/index.php?course=10516
- Randfälle beachten!
- Rekursionsbaum schön übersichtlich?!
- Aufgabe 1 e) : Rekurrenz muss wieder 2^n ergeben

Zweite Vereinfachung: Asymptotik

$$O(f(n)) = \{g(n) : \exists c > 0 : \exists n_0 \in \mathbb{N}_+ : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\}$$

„höchstens“

$$\Omega(f(n)) = \{g(n) : \exists c > 0 : \exists n_0 \in \mathbb{N}_+ : \forall n \geq n_0 : g(n) \geq c \cdot f(n)\}$$

„mindestens“

$$\Theta(f(n)) = O(f(n)) \cap \Omega(f(n))$$

„genau“

$$o(f(n)) = \{g(n) : \forall c > 0 : \exists n_0 \in \mathbb{N}_+ : \forall n \geq n_0 : g(n) \leq c \cdot f(n)\}$$

„weniger“

$$\omega(f(n)) = \{g(n) : \forall c > 0 : \exists n_0 \in \mathbb{N}_+ : \forall n \geq n_0 : g(n) \geq c \cdot f(n)\}$$

„mehr“

Aufgaben zum O-Kalkül

Für jedes der folgenden Paare von Funktionen gilt entweder $f(n) = O(g(n)$, $f(n) = \Omega(g(n)$ oder $f(n) = \Theta(g(n))$. Geben Sie an, welche Beziehung gilt und beweisen Sie deren Gültigkeit:

1. $f(n) = \log n^2$; $g(n) = \log n + 5$
2. $f(n) = \sqrt{n}$; $g(n) = \log n^2$
3. $f(n) = n \log n + n$; $g(n) = \log n$
4. $f(n) = n$; $g(n) = \log^2 n$

Aufgaben zum O-Kalkül

Für jedes der folgenden Paare von Funktionen gilt entweder $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$ oder beides. Geben Sie an, welche Beziehung gilt und beweisen Sie deren Gültigkeit:

1. $f(n) = 4n \log n + n$; $g(n) = (n^2 - n)/2$
2. $f(n) = n + \log n$; $g(n) = \sqrt{n}$
3. $f(n) = (n^2 - n)/2$; $g(n) = 6n$

Merging-Problem

Gegeben: zwei aufsteigend sortierte Arrays $A[1..n_1], B[1..n_2]$ von natürlichen Zahlen
Gesucht: das aufsteigend sortierte Array $C[1..(n_1 + n_2) =: n]$ von natürlichen Zahlen, das genau die Zahlen von A und B enthält

Algorithmus

procedure *merge*$(A$: Array $[1..n_1]$ of $_{\geq 0}$, B : Array $[1..n_2]$ of $_{\geq 0})$
$A[n_1+1] := \infty,\ B[n_2+1] := \infty$
$n := n_1 + n_2$
$j_A := 1,\ j_B := 1;$
for $i := 1$ **to** n **do**
  $C[i] = \min(A[j_A], B[j_B])$
  if $A[j_A] < B[j_B]$ **then**
    $j_A = j_A + 1$
  else
    $j_B = j_B + 1$
postcondition $C[i] \leq C[j] \quad \forall i \leq j,\ i,j \in \{1, ..., n\}$
postcondition $C[1..n]$ enthält genau $A[1..n_1], B[1..n_2]$
return C