Theoretische Informatik 3

1. Übung

Abgabe der schriftlichen Lösungen bis zum 6. Mai 2008

Für einen Übungsschein sind folgende Kriterien zu erfüllen:

- Lösen von $\geq 50\%$ der schriftlichen Aufgaben und
- Ankreuzen von $\geq 50\%$ der mündlichen Aufgaben.

Die schriftlichen Aufgaben können in Gruppen von maximal drei Personen bearbeitet und abgegeben werden. Diese Gruppen sollen über das ganze Semester gleich bleiben.

Aufgabe 1 [*mündlich*]

Für eine Reihe von algorithmischen Problemstellungen wurden 6 verschiedene Algorithmen mit folgenden Laufzeiten entworfen ($\log n$ steht als Abkürzung für $\lceil \log_2 n \rceil$):

Algorithmus	$A1$	$A2$	$A3$	$A4$	$A5$	$A6$
Laufzeit	$5 \cdot 10^8 n$	$10^5 n \log n$	$10^3 n^2$	$10 \cdot 2^{n/2}$	2^{2n}	$n!$

Die Algorithmen werden auf einem Rechner implementiert, der mit einer Geschwindigkeit von 10^9 Operationen pro Sekunde arbeitet.

a) Bestimmen Sie jeweils die maximale Länge der Probleminstanzen, die mit diesen Algorithmen innerhalb einer 1 Minute lösbar sind.

b) Um wieviel vergrößert sich jeweils die maximale Eingabelänge, wenn ein Rechner mit 10-facher Geschwindigkeit benutzt wird?

Aufgabe 2 [*5 Punkte*]

Seien $f, g : \mathbb{N} \to \mathbb{R}^+$ Funktionen. Beweisen oder widerlegen Sie folgende Aussagen:

a) $\sum_{i=1}^{n} i = O(n^2)$, (*mündlich*)

b) $\sum_{i=1}^{n} i^2 + \log n^9 = \Theta(n^3)$, (*mündlich*)

c) $\log^3 n = \omega(\log n^3)$, (*mündlich*)

d) $\log n^5 = \omega(\log n)$, (*1 Punkt*)

e) $f(n) + g(n) = O(\max\{f(n), g(n)\})$, (*mündlich*)

f) $O(f(n) + g(n)) = f(n) + O(g(n))$, (*mündlich*)

g) $f(n) + g(n) = \Omega(\min\{f(n), g(n)\})$, (*mündlich*)

h) $f(n) + g(n) = \Theta(\max\{f(n), g(n)\})$, (*mündlich*)

i) $\max\{f(n), g(n)\} = \Theta(f(n) + g(n))$, (*1 Punkt*)

j) $2^{n+1000} = O(2^n)$, (*mündlich*)

k) $2^{O(n)} = O(2^n)$, (*1 Punkt*)

l) $O(2^n) = o(3^n)$, (*mündlich*)

m) $O(2^n) = 3^{o(n)}$, (*1 Punkt*)

n) Wenn $f(n) = \Theta(g(n))$, dann gilt $f^2(n) = \Theta(g^2(n))$, (*mündlich*)

o) Wenn $f(n) = \Theta(g(n))$, dann gilt $f(n^2) = \Theta(g(n^2))$, (*mündlich*)

p) Wenn $f(n) = \Theta(g(n))$, dann gilt $\log f(n) = \Theta(\log g(n))$. (*1 Punkt*)

Aufgabe 3 [*mündlich*]

Für einen String $y = y_1 \cdots y_m \in \Sigma^*$ sei $L = \{x \in \Sigma^* \mid y \text{ ist Suffix von } x\}$. Zeigen Sie

$$\forall k = 0, \ldots, m : \ x \, R_L \, y_1 \cdots y_k \Leftrightarrow \max\{j \leq m \mid y_1 \cdots y_j \text{ ist Suffix von } x\} = k.$$

Aufgabe 4 [*mündlich*]

Geben Sie Ablaufprotokolle der beiden Algorithmen `DFA-String-Matcher` und `KMP-String-Matcher` bei Eingabe des Musters AUGAUGUAG und des Textes ACGAUGAUGUAGGCGAUGAUGUAG an.

Aufgabe 5 [*mündlich*]

Ein String x ist eine *zyklische Verschiebung* von y, falls $x \in \{uv \mid y = vu\}$ ist. Geben Sie einen Linearzeit-Algorithmus an, der für zwei Texte x und y entscheidet, ob x eine zyklische Verschiebung von y ist. Begründen Sie.

Aufgabe 6 [*mündlich*]

Sei $k \in \mathbb{N}$. Ein String $x = x_1 \cdots x_n$, $n \geq 1$, heißt *k-periodisch*, falls $x_i = x_{i+k}$ für alle $i = 1, \ldots, n-k$ gilt. Die Zahl $P(x) = \min\{k \geq 1 \mid x \text{ ist } k\text{-periodisch}\}$ heißt die *Periode* von x.

a) Geben Sie einen Linearzeit-Algorithmus an, der $P(x)$ berechnet. Begründen Sie.

b) Zeigen Sie, dass $P(x)$ genau dann ein von $|x| = n$ verschiedener Teiler von n ist, wenn es ein Wort y und eine Zahl $l > 1$ gibt mit $x = y^l$.

Aufgabe 7 [*5 Punkte*]

Ein String x heißt *zyklische Überdeckung* von y, falls y ein Teilwort eines Wortes in $\{x\}^*$ ist. Geben Sie einen Linearzeit-Algorithmus an, der für zwei Texte x und y entscheidet, ob x eine zyklische Überdeckung von y ist. Begründen Sie.