Institut für Informatik
Prof. Dr. Heiko Röglin
Clemens Rösner

universitätbonn

Algorithmen und Berechnungskomplexität II
Sommersemester 2014
Besprechung: 16. KW in den Übungen

Präsenzblatt

Aufgabe 1

Sei $G = (V, E)$ ein ungerichteter Graph mit $V = \{0, \ldots, n-1\}$. Beschreiben Sie eine Kodierung von G

(a) in Adjazenzmatrix-Darstellung über dem Alphabet $\Sigma = \{0, 1\}$,

(b) in Adjazenzlisten-Darstellung über dem Alphabet $\Sigma = \{0, 1, \#\}$,

(c) in Adjazenzlisten-Darstellung über dem Alphabet $\Sigma = \{0, 1\}$.

Aufgabe 2

Geben Sie die formale Darstellung des Sortierproblems als Relation und als Funktion an. Machen Sie sich dabei insbesondere Gedanken zur Kodierung der Eingabe und zum Alphabet. Beim Sortierproblem soll eine gegebene Menge M von natürlichen Zahlen in aufsteigend sortierter Reihenfolge wiedergegeben werden.

Aufgabe 3

Wir betrachten eine Grammatik $G = (\Sigma, V, S, P)$.

(a) Zeigen Sie, dass es eine Turingmaschine gibt, die die Sprache $L(G)$ entscheidet, wenn P nur Regeln der Form $\alpha \to \beta$ mit $\alpha, \beta \in (V \cup \Sigma)^*$ und $|\beta| > |\alpha|$ und Regeln der Form $A \to a$ für $A \in V$ und $a \in \Sigma$ enthält.

(b) Kann mit der Methode aus Aufgabenteil a) auch eine Turingmaschine für die Sprache $L(G)$ entworfen werden, wenn keine Einschränkungen an die Regeln der Grammatik gestellt werden?

Aufgabe 4

Wir betrachten zwei Probleme A und B. Problem A lässt sich durch folgenden Algorithmus lösen:

1. Berechne aus einer Eingabe x der Länge n für Problem A in Zeit $O(n^3)$ eine Eingabe y der Länge $N = O(n^3)$ für Problem B.
2. Rufe einen Algorithmus für Problem B mit Eingabe y auf.
3. Gib die Ausgabe dieses Algorithmus als Ausgabe für Problem A aus.

Die Komplexität eines Problems ist $O(f(n))$, wenn es einen Algorithmus für das Problem mit Laufzeit $O(f(n))$ gibt. Gibt es keinen Algorithmus mit Laufzeit $o(f(n))$, so hat das Problem Komplexität $\Omega(f(n))$.

Beantworten Sie die folgenden Fragen und begründen Sie Ihre Antworten:

(a) Was kann man über die Komplexität von Problem A aussagen, wenn Problem B Komplexität $O(N^2)$ hat, wobei N die Länge einer Eingabe für Problem B ist?

(b) Was lässt sich über die Komplexität von Problem A schlussfolgern, wenn wir wissen, dass es keinen Algorithmus für Problem B mit Laufzeit $O\left(e^N\right)$ gibt?

(c) Was lässt sich über die Komplexität von Problem B aussagen, wenn wir einen Algorithmus für Problem A mit Laufzeit $O(n)$ kennen?

(d) Was kann man über die Komplexität von Problem B schlussfolgern, wenn die Komplexität von Problem A $\Omega\left(e^n\right)$ ist?

(e) Was kann man über die Komplexität von Problem B schlussfolgern, wenn die Komplexität von Problem A $\Omega(n^2)$ ist?