# Übungsblatt 1

***Abgabe der schriftlichen Lösungen bis 30. Oktober 2019***

## Aufgabe 1 — *mündlich*

Betrachten Sie eine TSP-Instanz mit $n$ Städten und Entfernungen $d_{ij} \geq 0$. Wir definieren für jede Menge $S$ von Städten (mit $1 \notin S$) und für jedes $j \in S$ den Wert $c[S, j]$ als den kürzesten Weg von 1 nach $j$, der jede Stadt in $S \cup \{1\}$ genau einmal besucht.

(a) Geben Sie einen Algorithmus an, der alle $c[S, j]$ mit *dynamischer Programmierung* berechnet, also von kleinen auf große Stadtmengen $S$ schließt.

(b) Benutzen Sie diesen Algorithmus, um das TSP in Zeit $\mathcal{O}(n^2 2^n)$ zu lösen. Welchen Platzbedarf hat der resultierende Algorithmus?

## Aufgabe 2 — *mündlich*

Eine Turingmaschine heißt **blind** (engl. *oblivious*: vergesslich, blind), falls ihre Kopfpositionen zu jedem Zeitpunkt $t$ der Rechnung nur von $t$ abhängen.

(a) Zeigen Sie, dass jede Turingmaschine $M$ von einer blinden Turingmaschine $M'$ simuliert werden kann.

(b) Geben Sie eine obere Schranke für die Rechenzeit $\mathrm{time}_{M'}(x)$ von $M'$ in Abhängigkeit von $\mathrm{time}_M(x)$ an.

## Aufgabe 3 — *mündlich*

Betrachten Sie eine Turingmaschine $M$, die ein *zweidimensionales* Band zur Verfügung hat. Der Schreib-Lesekopf kann sich also auch nach oben und unten bewegen.

(a) Von welcher Form ist die Überführungsfunktion von $M$?

(b) Zeigen Sie, wie eine solche Turingmaschine durch eine DTM $M'$ simuliert werden kann.

## Aufgabe 4 — *mündlich*

Zeigen Sie: Jede $t(n)$-zeitbeschränkte $k$-DTM $M$ kann von einer *1*-DTM $M'$ in Zeit $\mathcal{O}(t(n)^2)$ simuliert werden. Überlegen Sie, wie sich die Simulation bei Verwendung einer *2*-DTM noch effizienter gestalten ließe?

## Aufgabe 5 — *mündlich*

Zeigen Sie: Jede Sprache, die von einer $k$-NTM $N$ in $f(n)$ vielen Schritten entschieden wird, kann auch von einer *2*-NTM $N'$ in $\mathcal{O}(f(n))$ vielen Schritten entschieden werden.

## Aufgabe 6 — *10 Punkte*

Bestimmen Sie für alle Paare $f_i, f_j$ der Funktionen

$$f_1(n) = n^2, f_2(n) = n^3, f_3(n) = n^2 \log n, f_4(n) = 2^n, f_5(n) = n^n,$$

$$f_6(n) = n^{\log n}, f_7(n) = 2^{2^n}, f_8(n) = 2^{2^{n+1}} \text{ und } f_9(n) = \begin{cases} n^2 & n \text{ gerade}, \\ 2^n & n \text{ sonst}, \end{cases}$$

ob

(a) $f_i(n) \in O(f_j(n))$,
(b) $f_i(n) \in \Omega(f_j(n))$, oder
(c) $f_i(n) \in \Theta(f_j(n))$ gilt.