Theoretische Informatik 1 / Algorithmentheorie

Wintersemester 2012/13

Prof. Dr. Georg Schnitger,
Dipl. Inf. Bert Besser,
Dipl. Inf. Matthias Poloczek

Arbeitsgruppe Theoretische Informatik, Institut für Informatik

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Blatt 7

Ausgabe: 13.12.2012
Abgabe: 20.12.2012 **vor** der Vorlesung

7.1. Aufgabe (3) *Maximale aufsteigende Teilfolgen*

Wir betrachten das Problem der maximalen aufsteigenden Teilfolge, das wir bereits in Aufgabe 6.2 kennengelernt haben. Gegeben ist eine Folge von Zahlen $a_1, a_2, \ldots, a_n$. Gesucht ist eine aufsteigende Teilfolge $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ maximaler Länge. Eine Teilfolge $(i_1 < i_2 < \cdots < i_k)$ ist aufsteigend, wenn $a_{i_1} < a_{i_2} < \cdots < a_{i_k}$ gilt.

Zur Lösung des Problems soll dynamische Programmierung verwendet werden. Dabei beschreibt Teilproblem $TF[i]$ die Länge einer längsten aufsteigenden Teilfolge, die mit a_i endet.

Bestimme für die Eingabe

$$11, 6, 3, 8, 12, 10, 5, 7, 4, 9$$

die Lösungen der Teilprobleme $TF[i]$.

Allerdings ist nicht nur die Länge einer maximalen aufsteigenden Teilfolge zu berechnen, sondern auch diese selbst zu bestimmen. Welche Informationen musst Du für jedes Teilproblem speichern, um die optimale Löung zu rekonstruieren? **Führe** Deinen Ansatz für diese Eingabe aus.

7.2. Aufgabe (3) *Die polynomielle Reduktion*

Betrachten wir zwei Probleme A und B. Wir wissen, dass Problem A NP-hart ist. Zusammen mit der Annahme $\mathsf{P} \neq \mathsf{NP}$ folgt daraus, dass für A kein deterministischer Algorithmus existiert, der das Problem *effizient*, d.h. in polynomieller Zeit löst.

Gib eine informelle Erklärung, warum aus der Reduktion $A \leq_p B$ folgt, dass B ebenfalls „schwierig“ ist. Oder muss man für diesen Schluss die Reduktion $B \leq_p A$ nutzen? Warum?

Und warum ist es so wichtig, dass die transformierende Turingmaschine in polynomieller Zeit arbeitet?

7.3. Aufgabe (9) *Polynomielle Algorithmen*

Wir geben vier Probleme an, für die ein effizienter Algorithmus gesucht werden soll. Wir sagen, dass ein Algorithmus effizient ist, wenn seine Laufzeit polynomiell in der Länge der Eingabe ist. Wenn also ein Entscheidungsproblem Q einen effizienten Algorithmus besitzt, dann gehört Q zur Klasse P.

Es genügt, die Algorithmen in Pseudo Code zu beschreiben und die polynomielle Laufzeit nachzuweisen.

Gib für drei der genannten Probleme polynomielle Algorithmen an.

a.) Gegeben ist ein String a^n. Es soll entschieden werden, ob n eine Primzahl ist.

b.) Gegeben ist eine Zahl n in Binärdarstellung. Es soll entschieden werden, ob n das Quadrat einer ganzen Zahl ist. *Beachte, dass die Eingabelänge nicht n selbst, sondern die Länge seiner Binärdarstellung ist.*

Bei einer legalen Knotenfärbung wird jedem Knoten eine Farbe zugewiesen. Benachbarte Knoten müssen dabei verschiedene Farben erhalten.

c.) Gegeben ist ein ungerichteter Graph $G = (V, E)$. Es soll entschieden werden, ob es möglich ist, G mit höchstens 3 Farben legal zu färben.

d.) Gegeben ist ein ungerichteter Graph $G = (V, E)$. Es soll entschieden werden, ob es möglich ist, G mit höchstens 2 Farben legal zu färben.

7.4. Aufgabe (9) *Set Packing*

Im Rahmen dieser Vorlesung beschränken wir uns weitgehend auf Entscheidungsvarianten von Problemen; in praktischen Anwendungen jedoch sind Entscheidungs– und Optimierungsvarianten oftmals mit einem vergleichbaren Aufwand lösbar.

Wir betrachten in dieser Aufgabe drei verschiedene Varianten des Problems *SET PACKING* (SP): Sei $C = \{C_1, \ldots, C_m\}$ eine Menge endlicher Mengen.

Variante 1: Enthält C mindestens k disjunkte Teilmengen? Hierbei ist die natürliche Zahl k Teil der Eingabe.

Variante 2: Berechne die maximale Anzahl disjunkter Teilmengen aus C.

Variante 3: Gib eine größtmögliche Menge von disjunkten Teilmengen aus C aus.

Die erste Variante heißt auch Entscheidungsvariante. Die zweite Variante fragt nach dem Wert einer optimalen Lösung. Die dritte Variante erfordert die Berechnung einer optimalen Lösung.

a.) Angenommen Variante 1 ist effizient entscheidbar. Ist dann auch Variante 2 deterministisch in Polynomialzeit lösbar? **Begründe** Deine Antwort.

b.) Angenommen Variante 1 ist effizient entscheidbar. Ist dann auch Variante 3 deterministisch in Polynomialzeit lösbar? **Begründe** Deine Antwort.