Abt. Formale Konzepte	Universität Stuttgart
Fakultät Informatik	SS 01, 29.06.2001
Abgabe bis Mittwoch 04.07.2001	Zugehörige Übungen am 06.07.2001 8.30 Uhr
Empfohlene Bearbeitungszeit:3 Stunden	Hierbei mindestens zu erreichende Punktzahl: 12.

Übung 5

zu Theoretische Informatik III (für Softwaretechnik)

Aufgabe 1: Quadratteile (4 Punkte)

Sei Σ ein Alphabet, $u, w, x, y \in \Sigma^*$. u heißt *Quadratteil* von $w \Leftrightarrow w = xuuy$. Z.B. kann das Wort $aabab$ in $x = \varepsilon$, $u = a$ und $y = bab$ zerlegt werden. w heißt *quadratfrei* $\Leftrightarrow$ es existiert kein Quadratteil in w. Z.B. ist das Wort aba quadratfrei.

Konstruieren Sie eine nicht-deterministische Turingmaschine, die in linearer Zeit entscheidet, ob ein gegebenes Wort w einen Quadratteil hat.

Aufgabe 2: TSP – Nicht-Determinismus (3 Punkte)

Gegeben sei ein Graph $G = (V, E, \delta)$ und eine Zahl m. Zeigen Sie: Das Problem, ob es in diesem Graphen einen Weg w mit $\delta(w) \leq m$ gibt, der alle Knoten enthält, kann nicht-deterministisch in polynomieller Zeit entschieden werden (liegt in **NP**).

Aufgabe 3: Bucketsort (6 Punkte)

Gegeben sei ein Array R of integer (Register). Im Register $R[0]$ ist der Wert n gespeichert. In den nächsten n Stellen sind n Zahlen vorgegeben (die sortiert werden sollen). Schreiben Sie ein Programm (keine Registermaschine), welches folgenden Algorithmus nachvollzieht:

Phase 1: Verteilen der Werte auf das Intervall $[min, max]$.
In $R[n+1]$ wird der minimale Wert ($= min$) und in $R[n+2]$ der maximale Wert ($= max$) von $R[1]$ .. $R[n]$ gespeichert. In $R[n+3]$ wird gezählt, wie oft der Wert min in $R[1]$ .. $R[n]$ vorkommt. In $R[n+4]$ wird gezählt, wie oft der Wert $min+1$ in $R[1]$ .. $R[n]$ vorkommt. usw.

Phase 2: Zurückschreiben der Werte (= sortierte Reihenfolge).
Sei r der Wert von $R[n+3]$. In die unteren r Register $R[1]$ .. $R[r]$ wird der Wert min geschreiben. Sei s der Wert von $R[n+4]$, dann wird in die nächsten s Register $R[r+1]$ .. $R[r+s]$ der Wert $min+1$ geschrieben. usw.

a) Formulieren Sie den Algorithmus aus.

b) Wann hat der Algorithmus lineare Laufzeit? Welches Laufzeitverhalten hat der Algorithmus allgemein.

Aufgabe 4: Reduktion (6 Punkte)

Gegeben sei ein Graph $G = (V, E, \delta)$ $(\delta : E \to \mathbb{N})$, $u, v \in V, d \in \mathbb{N}$.

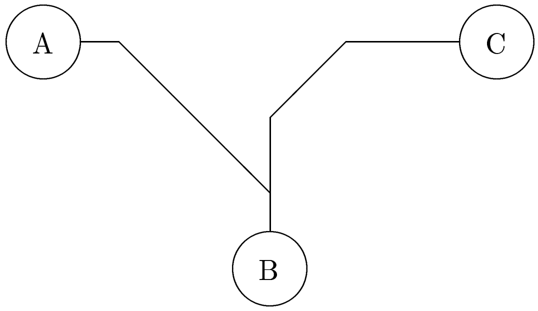
Problem 1: (LWeg) Existiert ein doppelpunktfreier Weg w von u nach v mit $\delta(w) = d$?

Problem 2: (Ham) Hat G einen Hamiltonschen Weg (= doppelpunktfreier Weg in welchem alle Knoten genau einmal vorkommen)?

Zeigen Sie: Ham $\leq_{pol}$ LWeg.

Aufgabe 5: Reduktion (6 Punkte)

Friedhelm hat Eisenbahngleise wie folgt aufgebaut (B wirkt als Keller zwischen A und C):



Im Bahnhof A stehen n Waggons. Friedhelm definiert 2 Operationen:

1) **Push**: Verschiebt den vorderen Waggon im Bahnhof A in den Bahnhof B.

2) **Pop**: verschiebt den letzten Waggon im Bahnhof B (den zuletzt in den Bahnhof B verlegten Waggon) in den Bahnhof C.

a) Speziell sei n =4. Welche verschiedenen Permutationen der Waggons sind im Bahnhof C möglich, wenn alle Waggons von A nach C verschoben wurden?

b) Zeigen Sie: Die Anzahl der Permutationen mit einem Keller (bei Eingabe von n Elementen) = Anzahl der korrekten Klammerungen für das Klammerpaar '(' , ')' mit n Klammerpaaren.

Aufgabe 6: P und **NP** (3 Punkte)

Sei $\{1\}$ das Problem zu entscheiden, ob ein Element x in der Menge $\{1\}$ enthalten ist oder nicht.

Zeigen Sie: $\{1\}$ ist **NP**-vollständig $\Leftrightarrow$ **P**=**NP**