Übungen zur Vorlesung

Komplexitätstheorie

Aufgabenblatt 9

In der Übung Dienstag 20.1.09 um 10.15 Uhr im HZ204
werden die Übungsaufgaben vorgerechnet.

Aufgabe 1

Das Problem MD3VC sei:
Gegeben: $G(V, E)$, k und für alle $v \in V$ haben wir $d(v) \leq 3$.
Gesucht: Gibt es ein Vertex Cover mit maximal k Knoten.

Zeigen Sie, dass MD3VC NP-hart ist.

Tipp: Studieren Sie nochmals die Reduktion von 3-SAT auf VERTEX COVER. Überlegen Sie, welche 3-SAT-Variante Sie auf MD3VC reduzieren müssen.

Aufgabe 2

Formuliere folgende Probleme als ILP:

1. WEIGHTED INDEPENDENT SET.
2. WEIGHTED DOMINATING SET.
3. MINIMUM MAKESPAN
4. 3-SAT

Überlegen Sie sich, dass eine Gleichung der Form $a \leq b$ in eine äquivalente Gleichung $a' \geq b'$ umgeformt werden kann.
Ebenso ist es möglich, wenn eine lineare Funktion f maximiert wird sie durch ein f' zu ersetzen welches minimiert wird ohne den zulässigen Lösungsraum zu verkleinern oder vergrössern.

Aufgabe 3

VERTEX COVER NUMBER ist:
Gegeben: $G(V, E)$ und k.
Frage: Ist k die kleinste Zahl, sodass G ein Vertex Cover der Größe k besitzt?

Zeigen Sie:

1. VERTEX COVER NUMBER ist NP-hart.
2. VERTEX COVER NUMBER liegt in PSPACE.

3. Vertex Cover Number $\in$ P $\iff$ P = NP.

4. Vertex Cover Number $\in$ NP $\iff$ co-NP = NP.

Aufgabe 4
Subset-Sum
Eingabe: $S = \{w_1, w_2, ..., w_n\}$, $w_i \in \mathbb{N}$ and $B \in \mathbb{N}$.
Ausgabe: Gibt es $I \subseteq S$ mit $\sum_{\ell \in I} w_\ell = B$.

Sei (j, w)-Subset-Sum
Eingabe: $S = \{w_1, w_2, ..., w_j\}$, $w_i \in \mathbb{N}$ and $1 \leq w \leq B$, $w \in \mathbb{N}$.
Ausgabe: Gibt es $I \subseteq S$ mit $\sum_{\ell \in I} w_\ell = B$.

Zeigen Sie

1. Sie können (j, w)–Subset-Sum lösen, falls Sie $(j-1, w)$-Subset-Sum und $(j-1, w-w_j)$-Subset-Sum bereits gelöst haben.

2. Sei $M[i, w] = 1 \iff (i, w)$-Subset-Sum hat eine Lösung. Sonst ist der Eintrag 0. Die Dimension der Matrix M sei $n \times B$.
   Schreiben Sie einen Algorithmus, der sukzessive M mit Hilfe der in 1. gezeigten Relation auffüllt (und somit $M[n, B]$ berechnet).

3. Können Sie einen Algorithmus finden mit Laufzeit $O(\sqrt{2}^n)$? (Tipp: Teilen Sie die Eingabe in zwei Hälften und betrachten Sie in beiden Hälften alle Möglichkeiten. Wie lassen diese sich dann kombinieren? Sortieren!)