Übungen zur Vorlesung **Wintersemester 2011/12**
Algorithmen für schwierige Probleme **Universität Ulm**
Britta Dorn **10.11.2011**
Jacobo Torán

Blatt 2

(Besprechung am 16.11.2011)

Aufgabe 1. (Problemkern für Independent Set **auf planaren Graphen)**

Das NP-vollständige Problem Independent Set ist wie folgt definiert:

Independent Set
Eingabe: Ein ungerichteter Graph $G = (V, E)$ sowie eine natürliche Zahl k.
Aufgabe: Finde eine Teilmenge $I \subseteq V$ bestehend aus k oder mehr Knoten, die ein Independent Set bilden, d.h., I induziert einen kantenlosen Teilgraphen von G.

Zeige unter Verwendung des nachfolgenden Satzes, dass das Problem Independent Set beschränkt auf planare Graphen einen Problemkern der Größe $4k$ besitzt. Warum ist diese Problemkernreduktion nicht wirklich befriedigend?

Satz: Jeder planare Graph $G = (V, E)$ lässt sich in Polynomzeit *vier-färben*, d.h. man findet eine Abbildung $\phi : V \to \{1, 2, 3, 4\}$ so dass gilt: $\{v, w\} \in E \Rightarrow \phi(v) \neq \phi(w)$.

Aufgabe 2. (Problemkern für Closest String**)**

Das Closest String-Problem wird unter anderem zum Entwurf von Primern für DNA-Replikation verwendet.

Closest String
Eingabe: Eine Menge von k gleich langen Zeichenketten $s_1, s_2, \ldots, s_k$ über einem Alphabet Σ, und eine positive ganze Zahl d.
Aufgabe: Finde eine Zeichenkette s, so dass $d_H(s, s_i) \leq d$ für jedes $i = 1, \ldots, k$.

Dabei ist $d_H(s, t)$ der Hamming-Abstand von zwei gleich langen Zeichenketten s und t, d. h. die Anzahl von Positionen, an denen sich s und t unterscheiden.
Zeige einen Problemkern mit höchstens $k \cdot d$ Spalten für Closest String.

Aufgabe 3. (Spezialfall bei Closest String**)**

Betrachte den Fall, dass es in einer Closest String-Instanz zwei Zeichenketten s_i, s_j mit $d_H(s_i, s_j) = 2d$ gibt. Wie kann man in diesem Fall das Problem schneller lösen? Zeige, dass hier ein Suchbaum der Größe $O(4^d)$ ausreicht.

Aufgabe 4. (Modellierung eines VLSI-Problems)

Beim „rekonfigurierbaren VLSI“ (VLSI: Very-large-scale integration) tritt folgendes Problem auf: Gegeben ein rechteckiges $m \times n$-Feld, in welchem einige der $m \cdot n$ Zellen fehlerhaft sein können. Gesucht wird nun eine minimale Anzahl von Zeilen und Spalten, die alle fehlerhaften Zellen abdecken. Parametrisiert ist dies die Frage, ob für gegebene Parameterwerte k_1 und k_2 alle Fehler mit maximal k_1 Zeilen und k_2 Spalten abgedeckt werden können.

Gebe einen Suchbaumalgorithmus mit Laufzeit $f(k_1, k_2) \cdot (m+n)^{O(1)}$ an und zeige damit, dass dieses Problem in FPT ist.