9. Übungsblatt zu Grundzüge von Algorithmen und Datenstrukturen, WS 15/16

Prof. Markus Bläser
`http://www-cc.cs.uni-saarland.de/course/50/`

Abgabe: 14. Januar 2016, 12:00 Uhr

Aufgabe 9.1 Der folgende Algorithmus ist ein Greedy-Algorithmus, der eine zulässige Knotenfärbung eines Graphs G berechnet:

Input: Graph $G = (V, E)$ mit Knoten $V = \{v_1, \ldots, v_n\}$ und ungerichteten Kanten $E \subseteq V \times V$

Output: Colors[1..n]

```
 1: Colors[1..n] ← {0, ..., 0}
 2: for i = 1...n do
 3:     Forbidden[1..n] ← {FALSE, ..., FALSE}
 4:     for j = 1...i − 1 do
 5:         if {v_i, v_j} ∈ E then
 6:             Forbidden[Colors[j]] ← TRUE
 7:     colorI ← 1
 8:     while Forbidden[colorI] = TRUE do
 9:         colorI ← colorI + 1
10:     Colors[i] ← colorI
```

In Zeile 3 - 10 ermitteln wir eine zulässige Färbung für den Subgraph mit den Knoten $v_1, \ldots, v_i$, wobei in Zeile 7 - 10 die kleinste "freie" Farbe dem Knoten v_i zugewiesen wird.

Finden Sie für jedes n einen Graphen mit n Knoten, für den der Greedy-Algorithmus keine optimale Lösung liefert. Man kann Graphen finden, für die der Algorithmus mindestens $\lfloor \frac{n-2}{2} \rfloor + \chi(G_n)$ Farben benötigt.

Aufgabe 9.2 Gegeben ist ein Array mit n ganzen Zahlen. (Die Größe der Zahlen spielt in der Aufgabe keine Rolle, nehmen Sie an, dass man die Zahlen in Zeit $O(1)$ zuweisen, addieren, etc. kann.) Es sei N die Summe aller Zahlen in dem Array. Beim sogenannten Partition-Problem ist gefragt, ob es eine Teilmenge der Zahlen gibt, die sich exakt zu $N/2$ summiert. Verwenden Sie dynamische Programmierung, um einen Algorithmus zu entwerfen, der das Partition-Problem in Zeit $O(nN)$ löst.

Das Partition-Problem ist NP-vollständig. Warum folgt hieraus nicht P = NP? *(0 Zusatzpunkte)*

Aufgabe 9.3 Ein Automat soll einen Geldbetrag N mit Münzen ausgeben. Dies soll mit möglichst wenig Münzen geschehen. Es wird folgende Greedy-Strategie verwendet: Es wird immer die Münze mit dem größten Wert m ausgewählt, so dass $m \leq N$ ist, und mit dem Restbetrag $N - m$ rekursiv fortgefahren.

a) Zeigen Sie, dass diese Strategie für Euros optimal ist.

b) Entwerfen Sie ein Münzsystem, für das der Greedy-Algorithmus nicht die optimale Lösung liefert.