Institut für Informatik I
Prof. Dr. Heiko Röglin
Tobias Brunsch

Algorithmen und Berechnungskomplexität III
Sommersemester 2011
Abgabe: 08.06.2011 in der Vorlesung

Übungsblatt 1

Aufgabe 1.1

3+3 Punkte

(a) Wir betrachten folgenden Algorithmus für die Optimierungsvariante von `VERTEX COVER`, bei der wir als Eingabe einen ungerichteten Graphen $G = (V, E)$ mit $n = |V|$ Knoten erhalten:

(1) Setze $X = \emptyset$.
(2) Wiederhole die Schritte (3) bis (5) solange G noch Kanten besitzt:
(3) Wähle einen Knoten v mit maximalem Grad.
(4) Setze $X = X \cup \{v\}$.
(5) Entferne v und alle zu v inzidenten Kanten aus G.
(6) Gib X aus.

Zeigen Sie, dass der Algorithmus einen Approximationsfaktor von $O(\log n)$ besitzt.

Hinweis: Nutzen Sie Wissen aus der Vorlesung.

(b) Zeigen Sie, dass der in der Vorlesung vorgestellte Greedy-Algorithmus für `SET COVER` auch dann einen Approximationsfaktor von $\Omega(\log n)$ hat, wenn wir uns auf Eingaben beschränken, bei denen alle Teilmengen Kosten 1 haben.

Aufgabe 1.2

2+2 Punkte

(a) Wie kann man aus einem minimalen Vertex Cover eines Graphen G effizient ein maximales Independent Set von G berechnen?

(b) In der Vorlesung haben Sie einen 2-Approximationsalgorithmus für `VERTEX COVER` kennengelernt. Welchen Approximationsfaktor können Sie mit der Erkenntnis aus Aufgabenteil (a) für `INDEPENDENT SET` folgern? Beweisen Sie Ihre Aussage.

Aufgabe 1.3

3+3 Punkte

(a) Geben Sie einen pseudopolynomiellen Algorithmus für das Rucksackproblem an, der in Zeit $O(n \cdot b)$ läuft.

Hinweis: Nutzen Sie dynamische Programmierung.

(b) Das *fraktionelle Rucksackproblem* ist eine Variante des Rucksackproblems, bei dem die Objekte nicht vollständig in den Rucksack gepackt werden müssen, d.h. das Problem kann wie folgt modelliert werden:

$$\begin{aligned} \textbf{Maximiere } & p_1 x_1 + \ldots + p_n x_n \\ \textbf{Nebenbedingungen: } & w_1 x_1 + \ldots + w_n x_n \le b\,, \\ & x_1, \ldots, x_n \in [0, 1]\,. \end{aligned}$$

Geben Sie einen Algorithmus an, der das fraktionelle Rucksackproblem in Polynomialzeit löst.

Aufgabe 1.4

3+2+3 Punkte

Ein Kassierer möchte Wechselgeld mit möglichst wenigen Münzen zurückzahlen. Die zur Verfügung stehenden Münzsorten seien gegeben durch $C = (c_1, \ldots, c_m)$ mit $1 = c_1 < c_2 < \ldots < c_m$. Wir nehmen an, dass von jeder Münzsorte hinreichend viele Münzen vorhanden sind. Somit können wir das Problem für einen gegebenen Wechselgeldbetrag $W \in \mathbb{N}$ formulieren als

$$\begin{aligned} \textbf{Minimiere } & \alpha_1 + \ldots + \alpha_m \\ \textbf{Nebenbedingungen: } & \alpha_1 c_1 + \ldots + \alpha_m c_m = W\,, \\ & \alpha_1, \ldots, \alpha_m \in \mathbb{N}_0 = \{0, 1, 2, \ldots\}\ . \end{aligned}$$

(a) In den Vereinigten Staaten gibt es folgende Münzen: Cent, Nickel, Dime und Quarter, d.h. $C_{\text{US}} = (1, 5, 10, 25)$. Geben Sie einen Greedy-Algorithmus an, der das Problem für das Münzsystem C_{US} und jeden Wechselgeldbetrag W löst.

(b) Geben Sie ein Münzsystem C und einen Wechselgeldbetrag W an, für die der von Ihnen angegebene Greedy-Algorithmus nicht funktioniert.

(c) Geben Sie einen Algorithmus an, der ein Münzsystem $C = (c_1, \ldots, c_m)$ und einen Wechselgeldbetrag W als Eingabe erhält und in Zeit $O(m \cdot W)$ eine optimale Lösung berechnet.