Theoretische Informatik 1 / Algorithmentheorie

Wintersemester 2012/13

Prof. Dr. Georg Schnitger,
Dipl. Inf. Bert Besser,
Dipl. Inf. Matthias Poloczek

Arbeitsgruppe Theoretische Informatik, Institut für Informatik

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Blatt 5

Ausgabe: 22.11.2012
Abgabe: 29.11.2012 **vor** der Vorlesung

5.1. Aufgabe (2+2+4) *Schwere und leichte Eingaben für Greedy*

Wir arbeiten in einer Wechselstube und möchten stets den von einem Kunden getauschten Betrag n mit einer möglichst kleinen Anzahl k von Scheinen und Münzen darstellen. Die Werte der Scheine und Münzen seien $1 = w_1 < w_2 < \cdots < w_l$.

Wir betrachten den folgenden Greedy-Algorithmus auf der Eingabe n: Wähle den größten Wert w_i, so dass $n' := n - w_i \geq 0$ und gebe dem Kunden die entsprechende Münze (bzw. den entsprechenden Schein); falls jetzt $n' \neq 0$, dann wiederhole mit Eingabe n'.

- Zeige, dass es es Folgen $1 = w_1 < w_2 < \cdots < w_l$ gibt, für die der Algorithmus nicht die kleinste Anzahl von Scheinen und Münzen ausgibt.
- Zeige, dass der Algorithmus selbst dann nicht immer die kleinste Anzahl an Münzen und Scheinen errechnet, wenn $w_i \geq 2w_{i-1}$ für $i \geq 2$ gilt.
- Zeige, dass der Algorithmus korrekt arbeitet, falls $w_i = 2w_{i-1}$ für $i \geq 2$ gilt.

5.2. Aufgabe (8) *Algorithmen für schwierige Eingaben*

Wir befinden uns noch immer im Szenario aus der vorhergehenden Aufgabe. Wir möchten einen mächtigeren Algorithmus entwickeln, der mit allen Eingaben umgehen kann.

Benutze die Entwurfsmethode des dynamischen Programmierens, um einen Algorithmus anzugeben, der für jede Wertefolge $1 = w_1 < w_2 < \cdots < w_l$ und für jede Eingabe n eine optimale Anzahl k von Scheinen und Münzen errechnet.

Beschreibe dazu

- zuerst die Teilprobleme, aus deren Lösungen Du eine optimale Lösung für das Gesamtproblem herleiten möchtest
- und dann die Rekursionsgleichung, die Du zur Lösung der Teilprobleme verwendest.

Gib dann den Algorithmus in Pseudocode an. Analysiere seine Laufzeit und beweise seine Korrektheit.

5.3. Aufgabe (8) *Teilmenge mit vorgegebener Summe*

Die *natürlichen* Zahlen $x_1, \ldots, x_n$ sowie die natürliche Zahl B sind gegeben. Es ist zu entscheiden, ob es eine Teilmenge $S \subseteq \{1, \ldots, n\}$ mit

$$\sum_{i \in S} x_i = B$$

gibt. Jedes x_i kann also nur einmal zu B beitragen. **Beschreibe** einen Algorithmus, der dieses Problem mit Hilfe der dynamischen Programmierung löst.

- Welche Teilprobleme benutzt Dein Algorithmus
- und wie sieht Deine Rekursionsgleichung aus?

Die Laufzeit $O(n \cdot B)$ kann erreicht werden.