Algorithmen und Datenstrukturen 2 **SoSe 2021**
Prof. Dr. Sándor Fekete **Abgabe:** 01.06.21
Matthias Konitzny **Rückgabe:** ab 07.06.21

Hausaufgabenblatt 2

Die Abgabe der Lösungen muss bis zum 01.06.21 um 23:59 Uhr erfolgen. Lösungen müssen per Mail in einer PDF-Datei (Dateiname „`blatt_[nr]_[name1]_[name2]_[grpnr].pdf`“) an die/den jeweilige*n Tutor*in geschickt werden. Email-Adressen sind unter `https://www.ibr.cs.tu-bs.de/alg/index.html` zu finden.

Hausaufgabe 1 (Wechselgeld): **(3+4+5 Punkte)**
Wir möchten einen Verkaufsautomaten programmieren, der das Wechselgeld mit möglichst wenig Münzen zurückgibt. Formal:

Gegeben: Eine Zahl $W \in \mathbb{N}$ und Münzwerte $1 = M_1 < M_2 < \cdots < M_m$.

Gesucht: Nicht-negative, ganze Zahlen $x_1, \ldots, x_m$, sodass

$$\sum_{i=1}^{m} x_i M_i = W \text{ und}$$

$$\sum_{i=1}^{m} x_i \text{ minimal.}$$

Algorithmus 1 zeigt einen Greedy-Algorithmus, der möglichst hochwertige Münzen priorisiert.

```
function CHANGE(W, M_1, ..., M_m)
    for i = m down to 1 do
        x_i := ⌊W / M_i⌋                ▷ Nimm Münze i so oft wie möglich.
        W := W − x_i · M_i
    return x_1, ..., x_m
```

Algorithmus 1: Ein Greedy-Algorithmus, der möglichst wenig Wechselgeld zurückgeben soll.

a) Algorithmus 1 ist tatsächlich optimal für übliche Währungssysteme (Euro, US-Dollar, etc.). Das Euro-Währungssystem besitzt dabei die folgenden Münztypen: 1-, 2-, 5-, 10-, 20-, 50-, 100- und 200-Cent-Münzen. Gibt Algorithmus 1 die minimale Anzahl an Münzen zurück, wenn 4-Cent-Münzen eingeführt werden? Begründe deine Antwort.

b) Sei nun $\text{OPT}(i, x)$ der minimale Wert, wie viele der erste i Münzen benötigt werden, um den Wert x zu erreichen. Gib eine Rekursionsgleichung an, die $\text{OPT}(i, x)$ für ein Währungssystem mit Münzen im Wert von $1 = M_1 < M_2 < \cdots < M_m$ bestimmt.

c) Zeige: Wenn $\frac{M_{i+1}}{M_i} \in \mathbb{N}$ für alle $1 \leq i < m$ gilt, dann ist der Algorithmus 1 optimal.
(Hinweis:
(1) Angenommen, Z sei der zu erreichende Wert und $Z > M_j$ für ein $2 \leq j \leq m$. Wie oft kann man jede Münze M_i mit $i < j$ maximal verwenden?
(2) Für eine Folge $a_1, \ldots, a_n$ gilt: $\sum_{i=1}^{n-1} a_{i+1} - a_i = a_n - a_1$)

Hausaufgabe 2 (SUBSET SUM): **(6+2 Punkte)**

a) Wende das dynamische Programm für SUBSET SUM aus der Vorlesung auf folgende Instanz an:

i	1	2	3	4	5
z_i	5	3	7	2	6

und $Z = 11$.

Fülle dazu die folgende Tabelle aus, wobei der Eintrag in Zeile i und Spalte x dem Wert $\mathcal{S}(x, i)$ entspricht.

i \ x	0	1	2	3	4	5	6	7	8	9	10	11
0												
1												
2												
3												
4												
5												

b) Wie kann das dynamische Programm für SUBSET SUM verwendet werden, um PARTITION zu lösen?