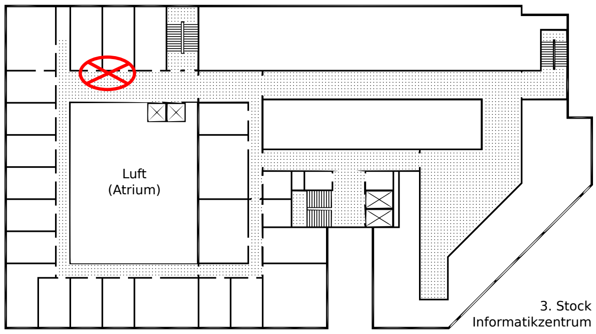
Algorithmen und Datenstrukturen II **SoSe 2019**
Dr. Linda Kleist **Abgabe:** 23.05.19
Arne Schmidt **Rückgabe:** 05./07.06.19

Übungsblatt 2

Abgabe der Lösungen bis zum 23.05.19 um 13:15 Uhr im Hausaufgabenschrank bei Raum IZ 337 (siehe Skizze rechts). Es werden nur mit einem dokumentenechten Stift (kein Rot!) geschriebene Lösungen gewertet. **Bitte die Blätter zusammenheften und vorne deutlich mit eigenem Namen, Matrikel- und Gruppennummer, sowie Studiengang versehen!**



Dieses Blatt besteht aus einer Präsenzaufgabe, die in der kleinen Übung besprochen wird, sowie aus zwei Hausaufgaben, die abgegeben werden müssen und bewertet werden.

Präsenzaufgabe:
Wir möchten einen Verkaufautomaten programmieren, der das Wechselgeld mit möglichst wenig Münzen zurückgibt.

a) Entwickle einen Greedy-Algorithmus, der für die Euro-Währung (d.h. mit 1-, 2-, 5-, 10-, 20-, 50-, 100- und 200-Cent-Münzen) die minimale Anzahl an Münzen zurückgibt.

b) Gibt der Algorithmus immer noch die minimale Anzahl an Münzen zurück, wenn 4-Cent-Münzen eingeführt werden? Begründe deine Antwort!

c) Entwickle ein dynamisches Programm, das für jedes Währungssystem mit Münzen im Wert von $1 = M_1 < M_2 < \cdots < M_m$ die minimale Anzahl an Münzen für das Wechselgeld zurückgibt.

d) Zeige: Wenn $\frac{M_{i+1}}{M_i} \in \mathbb{N}$ für alle $1 \leq i < m$ gilt, dann ist der Greedy-Algorithmus 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 1 (Subset Sum): **(6+2 Punkte)**

a) Wende das dynamische Programm für Subset Sum aus der Vorlesung auf folgende Instanz an:

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

und $Z = 12$

Fülle hierzu 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	12
0													
1													
2													
3													
4													
5													

b) Wie kann das dynamische Programm für Subset Sum verwendet werden, um Partition zu lösen?

Hausaufgabe 2 (Münzspiel): **(2+3+4+3 Punkte)**

In dieser Aufgabe betrachten wir folgendes Zwei-Spieler-Spiel: Auf einem Tisch liegen n Münzen im Wert von $M_1, \ldots, M_n$ in einer Reihe. Zwei Spieler, nennen wir sie Alice und Bob, ziehen nun abwechselnd eine Münze von einem der beiden Enden der Münzreihe. Sieger ist der Spieler, der am Ende einen größeren Gesamtmünzwert besitzt (bei Gleichstand gewinnt Alice). Beispiel:

$$2, 5, 1, 10, 50, 2, 2, 10$$

Alice wählt zunächst die (rechte) 10 und Bob danach eine zwei von rechts. Es bleibt über:

$$2, 5, 1, 10, 50, 2$$

Wählt Alice nun die rechte 2, kann Bob die Münze mit Wert 50 einstecken. Daher nimmt Alice die linke Münze. Da Bob die 50 nicht Alice überlassen möchte, wählt dieser die linke Münze. Es bleibt über:

$$1, 10, 50, 2$$

Die pfiffige Alice merkt, dass sie definitiv die 50 einstecken wird, wenn sie die 1 für sich beansprucht. Bob muss dann die 10 oder 2 einstecken, wodurch die 50 frei wird. Das Spiel neigt sich nach weiteren Ziehungen zu Ende und die beiden Spieler haben folgende Münzen gewählt haben:

Alice: 10, 2, 1, 50; Bob: 2, 5, 10, 2. Es steht 63 zu 19; ein klarer Sieg für Alice.

Alice sucht nun nach einer Strategie, mit der sie immer gewinnt, sofern es eine Möglichkeit gibt zu gewinnen. Wir nehmen an, dass Alice immer mit dem Ziehen beginnt.

a) Gib eine Instanz an (mit Begründung), bei der Alice nicht gewinnen kann.

b) Eine mögliche Gewinnstrategie wäre die folgende: Angenommen es liegen Münzen M_i bis M_j mit $i < j$ auf dem Tisch. Alice nimmt M_i falls $M_i - M_{i+1} \geq M_j - M_{j-1}$, und ansonsten M_j. Zeige, dass Alice mit dieser Strategie nicht immer gewinnt, obwohl sie gewinnen könnte. (Hinweis: Es darf angenommen werden, dass die Aussage in d) gilt.)

c) Sei $\text{OPT}(i, j)$ der optimale Wert, den ein Spieler erreichen kann, wenn die Münzen M_i bis M_j benutzt werden.

   (i) Konstruiere eine Rekursionsgleichung, die $\text{OPT}(i, j)$ berechnet. (Hinweis: Betrachte zusätzlich zu $\text{OPT}(i, j)$ den Münzwert $C_{i,j}$ der Münzen von i bis j. Von diesem Wert wird der Gegenspieler so wenig wie möglich übrig lassen!)

   (ii) Welchen Wert muss $\text{OPT}(1, n)$ überschreiten, damit Alice weiß, dass es eine Gewinnstrategie gibt?

d) Zeige: Liegen gerade viele Münzen auf dem Tisch, dann gibt es immer eine Gewinnstrategie für Alice.