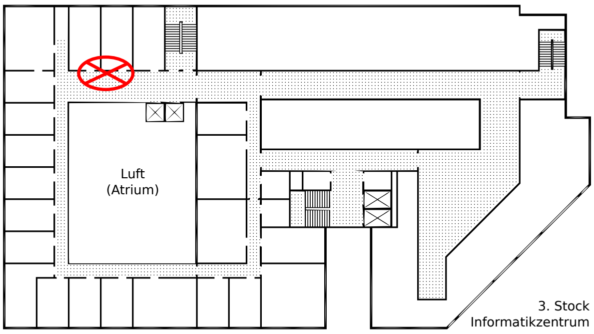
Algorithmen und Datenstrukturen II
Dr. Linda Kleist
Arne Schmidt

SoSe 2019
Abgabe: 09.05.19
Rückgabe: 15./17.05.19

Übungsblatt 1

Abgabe der Lösungen bis zum 09.05.19 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 mit Namen, Matrikel- und Gruppennummer 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:
Diese Aufgabe ist nur für die kleine Übung gedacht und soll nicht abgegeben werden.

a) Betrachte einen Brute-Force-Algorithmus für PARTITION, d.h. einen Algorithmus, der jede mögliche nicht-äquivalente Aufteilung der Elemente durchprobiert. Zwei Aufteilungen sind äquivalent, wenn die Elemente in identische Teilmengen aufgeteilt sind. Zum Beispiel ist eine Aufteilung von $\{1, 2, 3, 4\}$ in $S_1 = \{1, 2\}$ und $S_2 = \{3, 4\}$ äquivalent zur Aufteilung in $\tilde{S}_1 = \{3, 4\}$ und $\tilde{S}_2 = \{1, 2\}$.

(i) Gib alle nicht-äquivalenten Aufteilungen der folgenden drei Objekte an. Existiert für diese Instanz von PARTITION eine Lösung? Begründe Deine Antwort!

i	1	2	3	4
z_i	2	3	5	8

(ii) Zeige: Es gibt 2^{n-1} nicht-äquivalente Aufteilungen

b) Angenommen, wir haben einen Algorithmus $\mathcal{A}$, der 0-1-KNAPSACK löst, d.h. er gibt *true* zurück, falls es eine Auswahl der Objekte gibt, die in den Rucksack mit Kapazität Z passt und einen Wert P überschreitet. Andernfalls gibt $\mathcal{A}$ *false* zurück. Wie können wir $\mathcal{A}$ benutzen, um das Problem MAXIMUM KNAPSACK zu lösen? Welche Laufzeit besitzt dieser Algorithmus, wenn $\mathcal{A}$ die Laufzeit $\mathcal{T}(\mathcal{A})$ besitzt?

Hausaufgabe 1 (Fractional Knapsack): **(5+4 Punkte)**
In dieser Aufgabe betrachten wir Fractional Knapsack.

a) Sei $Z = 100$, und seien die sieben Objekte mit folgenden Werten gegeben:

i	1	2	3	4	5	6	7
z_i	24	26	20	20	20	20	16
p_i	10	13	8	12	15	5	5

Wende den Greedy-Algorithmus für Fractional Knapsack aus der Vorlesung auf diese Instanz an. Gib in jeder Iteration den aktuellen Gegenstand, den Anteil x_i zu dem er gepackt wird, sowie den Gesamtzustand (aktueller Gesamtwert und aktuelles Gesamtgewicht) an.

b) Sei I eine Instanz von Maximum Knapsack und P_{OPT} der optimale Lösungswert. Sei P_F der optimale Lösungswert, wenn I als eine Instanz von Fractional Knapsack interpretiert wird. Zeige: $P_{\text{OPT}} \leq P_F$

Hausaufgabe 2 (Maximum Knapsack): **(6+5 Punkte)**
In dieser Aufgabe betrachten wir Maximum Knapsack. Um dieses Problem zu lösen, wandeln wir den Algorithmus für Fractional Knapsack aus der Vorlesung folgendermaßen ab: Wir sortieren die Objekte aufsteigend nach $\frac{z_i}{p_i}$. Dann gehen wir die Objekte der Sortierung entsprechend durch und packen ein Objekt in den Rucksack, falls es hineinpasst. Eine formale Beschreibung ist in Algorithmus 1 zu sehen. Wir nennen diesen Algorithmus Greedy$_0$.

function Greedy$_0(z_1, \ldots, z_n, Z, p_1, \ldots, p_n)$
  Sortiere Objekte nach $\frac{z_i}{p_i}$ aufsteigend; dies ergibt Permutation $\pi(1), \ldots, \pi(n)$.
  for $j := 1$ **to** n **do**
    if $\left(\sum_{i=1}^{j-1} x_{\pi(i)} z_{\pi(i)} + z_{\pi(j)} \leq Z\right)$ **then**
      $x_{\pi(j)} := 1$
    else
      $x_{\pi(j)} := 0$
  return $x_1, \ldots, x_n$

Algorithmus 1: Greedy$_0$

a) Sei $Z = 23$, und seien die sechs Objekte mit folgenden Werten gegeben:

i	1	2	3	4	5
z_i	4	8	23	6	5
p_i	2	9	16	4	9

Wende Greedy$_0$ auf diese Instanz an. Entscheide in jeder Iteration, ob der aktuelle Gegenstand gepackt wird und gib den Gesamtzustand (aktueller Gesamtwert und aktuelles Gesamtgewicht) an. Ist die erhaltene Lösung optimal? Begründe deine Antwort!

b) Zeige: Die ganzzahlige Lösung P_G von Greedy$_0$ kann beliebig weit von einer optimalen Lösung P_{OPT} entfernt sein, d.h.: Für jedes $0 < c \leq 1$ gibt es eine Instanz, sodass $0 < \frac{P_G}{P_{OPT}} < c$.
(Hinweis: Es wird vorausgesetzt, dass $z_i < Z$ für alle $1 \leq i \leq n$ gilt.)