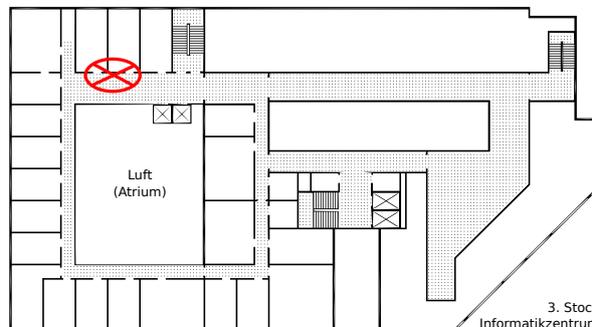


## Hausaufgabenblatt 4

Abgabe der Lösungen bis zum 06.07.22 um 12:00 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 beiden Namen, Matrikel- Übungs- und Gruppennummer versehen!**



### Hausaufgabe 1 (GREEDY<sub>k</sub>):

(8 Punkte)

In dieser Aufgabe betrachten wir den Algorithmus GREEDY<sub>k</sub> aus der Vorlesung. Wende GREEDY<sub>k</sub> auf die folgende Instanz an.

$i$	1	2	3	4	
$z_i$	7	9	11	9	mit $Z = 18$ und $k = 2$
$p_i$	9	9	10	8	

Gib dazu die folgenden Mengen bzw. Werte in jeder Iteration tabellarisch an:

- $\bar{S}$ : Menge fixierter Objekte
- $\sum_{i \in \bar{S}} z_i$ : Gewicht der fixierten Objekte
- $Z - \sum_{i \in \bar{S}} z_i$ : Restkapazität
- $G + \sum_{i \in \bar{S}} p_i$ : Wert der fixierten Objekte plus Greedy auf nicht fixierten Objekten.
- $G_k$ : Wert der bisher besten gefundenen Lösung
- $S$ : Lösungsmenge der bisher besten Lösung

Achte darauf, dass  $M$  mit einer kleinsten Menge anfängt und mit einer größten Menge endet. Zusätzlich soll  $M$  lexikographisch sortiert sein: Für zwei gleichgroße Mengen  $M_1$  und  $M_2$  kommt  $M_1$  vor  $M_2$ , falls das kleinste Element  $x \in M_1 \setminus M_2$  kleiner ist als das kleinste Element  $y \in M_2 \setminus M_1$ .

(Hinweis: Die Menge  $X \setminus Y$  enthält Elemente aus  $X$ , die nicht in  $Y$  vorkommen.)

### Hausaufgabe 2 (Wahlaufgabe):

(12 Punkte)

Im Folgenden kann zwischen einer theoretischen und einer praktischen Aufgabe gewählt werden. Es muss lediglich **eine** der Aufgaben bearbeitet werden, um die vollen Punkte zu erreichen. (Hinweis: Möchtest du dennoch beide Aufgaben bearbeiten, werten wir die Aufgabe mit der höheren Punktzahl.)

### Theoretische Aufgabe (Approximation mit GREEDY<sub>0</sub>) (3+1+1+3+4 Punkte)

Wir betrachten den Algorithmus GREEDY<sub>0</sub> für Instanzen mit folgenden Eigenschaften:

(A) Jedes Objekt besitzt ein Gewicht von höchstens  $\frac{Z}{\alpha}$  für ein festes  $\alpha \in \{1, 2, 3, 4, \dots\}$ .

(B) Die Summe aller  $z_i$  überschreitet die Kapazität  $Z$ .

Ohne Beschränkung der Allgemeinheit nehmen wir an, dass die Objekte bereits nach  $\frac{z_i}{p_i}$  aufsteigend sortiert sind.

- a) Sei das  $(k + 1)$ -ste Objekt das erste Objekt, welches GREEDY<sub>0</sub> nicht aufnimmt und sei OPT der optimale Lösungswert.

(i) Zeige:  $(\alpha - 1)p_{k+1} \leq \sum_{i=1}^k p_i$

(ii) Zeige:  $\sum_{i=1}^k p_i \leq \text{OPT}$

(iii) Zeige:  $\text{OPT} \leq \sum_{i=1}^{k+1} p_i$

- (iv) Zeige: GREEDY<sub>0</sub> liefert auf Instanzen mit Eigenschaften (A) und (B) eine  $(1 - \frac{1}{\alpha})$ -Approximation. (Hinweis: Nutze (i)-(iii).)

- b) Zeige: Der Approximationsfaktor ist bestmöglich. D.h., für jedes  $\alpha$  und für jedes  $c$  mit  $1 - \frac{1}{\alpha} < c \leq 1$  ist GREEDY<sub>0</sub> auf diesen Instanzen keine  $c$ -Approximation.

### Praktische Aufgabe (Branch-And-Bound Implementierung) (3+3+6)

In dieser Aufgabe soll der Branch-And-Bound Algorithmus für MAXIMUM KNAPSACK aus der Vorlesung in Python implementiert werden. Dazu soll GREEDY<sub>0</sub> (siehe Blatt 1) als untere Schranke und der (abgerundete Wert des) Greedy-Algorithmus für FRACTIONAL KNAPSACK als obere Schranke verwendet werden.

Nutze für die Entwicklung das Template unter [https://ibr.cs.tu-bs.de/courses/ss22/aud2/aufgaben/blatt4\\_template.py](https://ibr.cs.tu-bs.de/courses/ss22/aud2/aufgaben/blatt4_template.py). Dieses enthält auch eine Reihe von Beispielinstanzen mit denen die Implementierung überprüft werden kann.

Abgabe: Die Lösung dieser Aufgabe muss in Form einer Python-Datei (.py) an die/den jeweilige\*n Übungsleiter\*in gesendet werden. In der Mail muss der Name, die Matrikel-, Übungs- und Gruppennummer angegeben werden. Die jeweiligen Mailadressen gibt es unter <https://aud2.ibr.cs.tu-bs.de/index.php/organisation/>.

- a) Implementiere GREEDY<sub>0</sub>. Die Implementierung darf maximal 20 Zeilen lang sein.
- b) Implementiere den Greedy-Algorithmus für FRACTIONAL KNAPSACK. Die Implementierung darf maximal 20 Zeilen lang sein.
- c) Implementiere den Branch-And-Bound Algorithmus für MAXIMUM KNAPSACK aus der Vorlesung. Nutze dazu deine Implementierung aus a) und b), um die Schranken zu bestimmen. Die Implementierung darf maximal 50 Zeilen lang sein.