Technische Universität Braunschweig

Algorithmen und Datenstrukturen 2 – Übung #0

Matthias Konitzny
21.04.2021

Übersicht

- Organisation
- Wiederholung AuD 1
- Greedy für Fractional Knapsack

Organisation

Homepage

https://www.ibr.cs.tu-bs.de/courses/ss21/aud2/

Dort findet ihr

- Allgemeine Informationen
- Aktuelle Hinweise

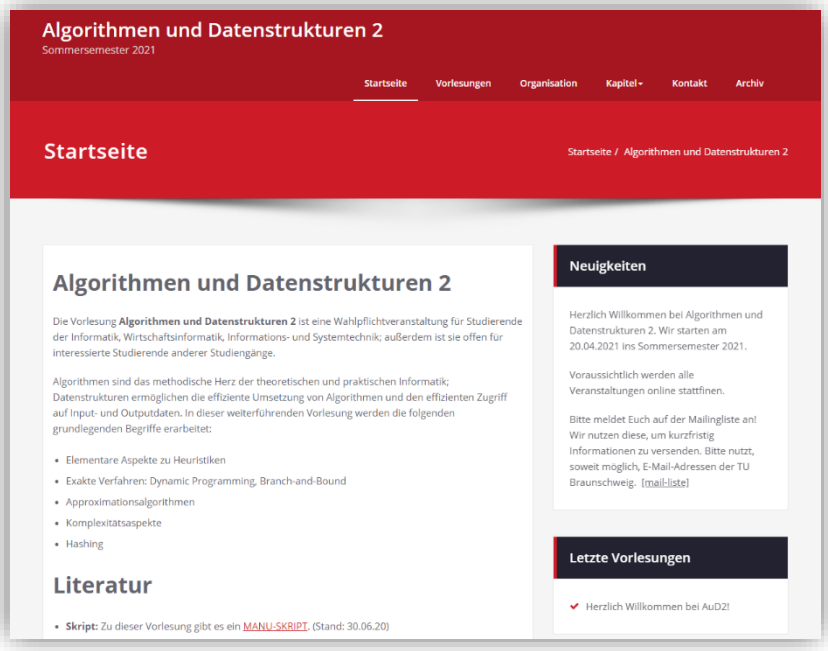
Organisation

Materialseite

https://aud2.ibr.cs.tu-bs.de

Dort findet ihr

- Aktuelle Informationen
- Vorlesungsvideos und Links zu den Live-Veranstaltungen
- Hausaufgaben
- Skript (bei Fehlern bitte Mail an mich)
- Literaturempfehlungen
- Weiterführende Links



Mailingliste

Mailingliste

Es gibt eine Mailingliste zu dieser Vorlesung. Bitte meldet euch mit eurer tu-bs-Email-Adresse dort an, denn wir werden sie nutzen, um kurzfristig Informationen zu verteilen. Bei technischen Schwierigkeiten wendet euch bitte an Matthias Konitzny.
Wichtig: Um Spam zu vermeiden, werden nur noch Mailadressen der TU Braunschweig freigegeben.

Anmeldung über Homepage

Information zur Vorlesung/Übung/Klausur/etc.

Bietet Möglichkeit Fragen zu stellen

- Wir (Prof. Fekete, Hiwis und ich) stehen auf der Liste und können Fragen beantworten
- Ihr könnt euch untereinander Fragen beantworten!

Subscribing to Aud2

Subscribe to Aud2 by filling out the following form. You will be sent email requesting confirmation, to prevent others from gratuitously subscribing you. Once confirmation is received, your request will be held for approval by the list moderator. You will be notified of the moderator's decision by email. This is also a private list, which means that the list of members is not available to non-members.

Your email address:	
Your name (optional):	

You may enter a privacy password below. This provides only mild security, but should prevent others from messing with your subscription. **Do not use a valuable password** as it will occasionally be emailed back to you in cleartext.

If you choose not to enter a password, one will be automatically generated for you, and it will be sent to you once you've confirmed your subscription. You can always request a mail-back of your password when you edit your personal options.

Pick a password:	
Reenter password to confirm:	
Which language do you prefer to display your messages?	English (USA)
Would you like to receive list mail batched in a daily digest?	◉ No ○ Yes

Subscribe

Anmeldung

Anmeldung zu kleinen Übungen auf der Homepage

Gruppe	Termin (14-täglich)	Tutor
01	Mittwoch: 09:45 – 11:15	Pia Meier
02	Mittwoch: 13:15 – 14:45	David Meyer
03	Freitag: 9:45 – 11:30	Anna Ronsdorf
04	Freitag: 13:15 – 14:45	Dennis Luck

Anmeldung

20. April 2021

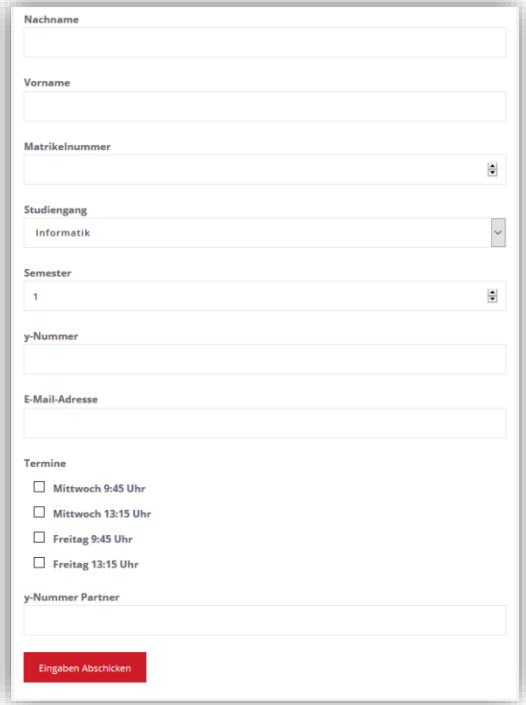
Anmeldung Kleine Übungen

in **Allgemein**

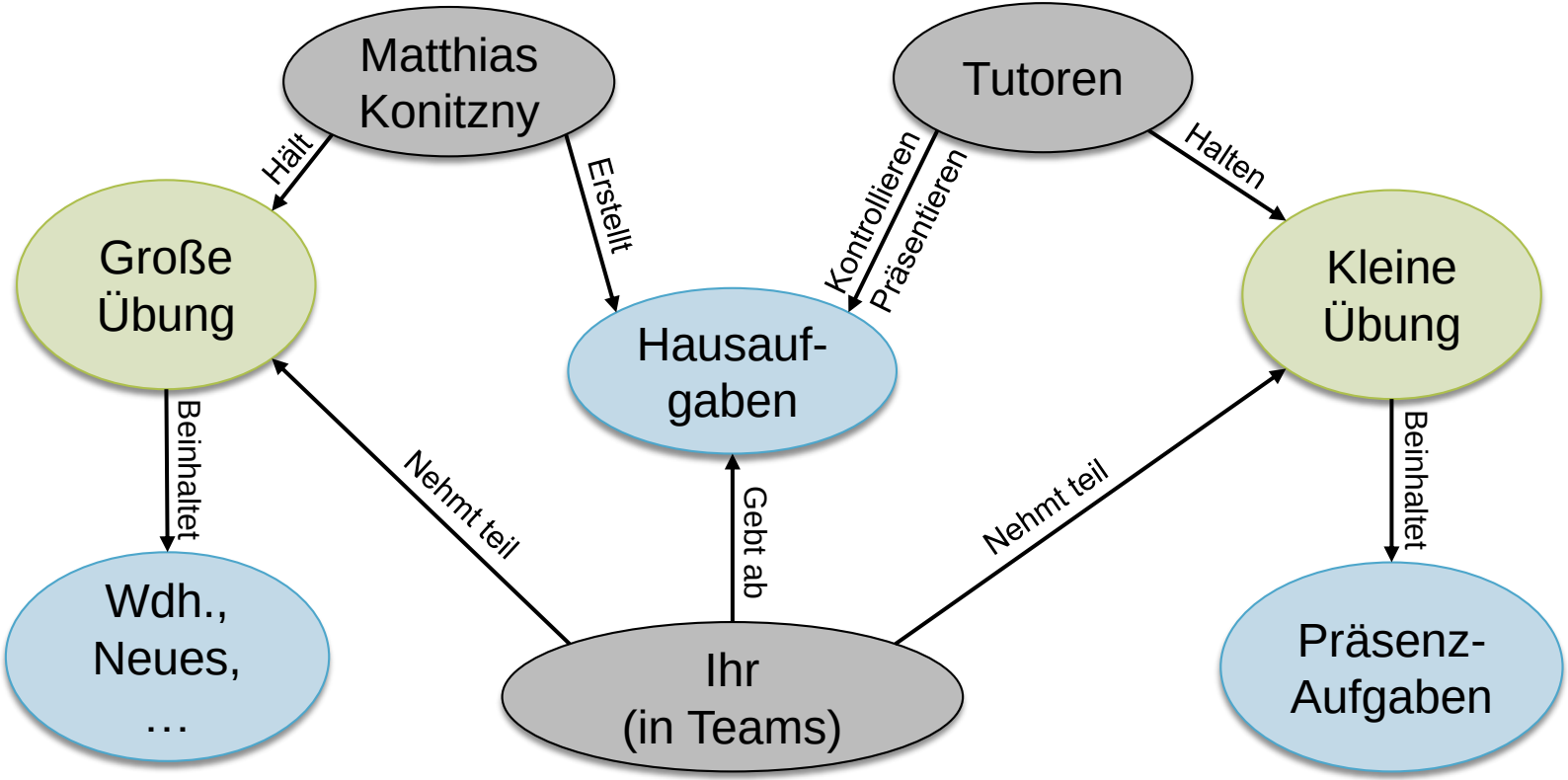
Die Anmeldung für die kleinen Übungen ist ab jetzt bis zum 26.04.21 geöffnet. Anschließend erhalten alle Teilnehmenden eine EMail mit der zugeteilten Übung und Gruppe.

Weitere Informationen

- Die Hausaufgaben sollen in Partnerarbeit erledigt werden. Dazu muss die y-Nummer des/der Wunschpartner*in im Formular angegeben werden.
- Die Wunschpartner*innen werden nur zugeteilt, wenn beide Partner*innen die y-Nummer des jeweils anderen angeben.
- Bitte achtet darauf, dass alle Angaben korrekt sind. Prüft dazu die Bestätigungsemail die ihr nach dem Anmeldevorgang erhaltet.
- Sollten mehrere Anmeldungen derselben Person vorliegen, berücksichtigen wir die Neueste.
- Sollten unvorhergesehene Probleme auftreten, meldet euch bei Matthias Konitzny



Übungsbetrieb



Hausaufgaben

- Voraussichtlich 5 bewertete Blätter à 20 Punkte → insgesamt 100 Punkte erreichbar.

- Für die Studienleistung müssen 50% der Punkte erreicht werden.
  - **ABER**: Keine Mindestpunktzahl pro Blatt!

- Bearbeitung in Zweiergruppen
  - Vorschlag: Jeder löst die Aufgaben selbstständig – anschließend Disskussion

- Abgabefrist steht auf dem jeweiligen Blatt.
- Zu spät abgegebene Hausaufgaben werden mit 0 Punkten bewertet
- Abgabe als Team per Mail an die Tutor*innen. (https://www.ibr.cs.tu-bs.de/alg/index.html)

Hausaufgaben abgeben

Wie werden Hausaufgaben abgegeben?

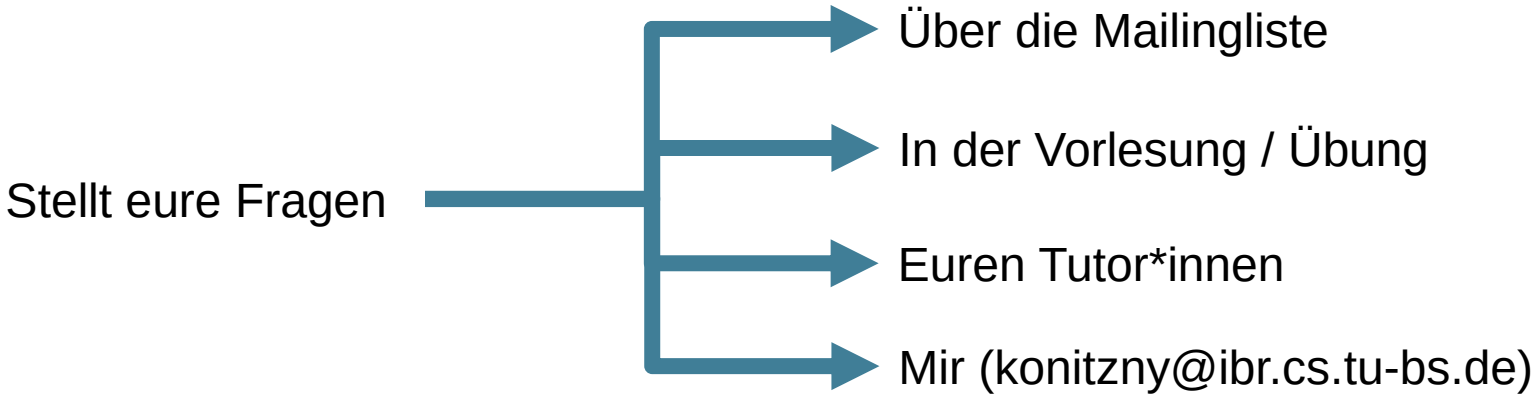
1. Erstellen der elektronischen Lösung
   - Einscannen/Abfotografieren einer auf Papier geschriebenen Lösung (ggf. große Dateien)
   - TeX, Word, OpenOffice, etc. (ggf. umständlich für Formeln)
   - Konvertierung ins PDF Format (nur eine einzige Datei!)
   - Dateiname im Format blatt[nr]_[name1]_[name2]_[gruppennummer].pdf

2. Abgabe per Mail
   - Eine Person schickt elektronische Lösung per Mail an den Tutor.
   - Alle Personen des Teams müssen auf CC stehen.
   - Nutzt keine Filehosting-Dienste (Dropbox, Google-Drive, etc)

Zeitlicher Ablauf

Anmeldung	Einteilung in Gruppen	0. kl. Übung	Einteilung in Teams
Bis zum 26.04.21 Um 23:59 Uhr	Am 27.04.21	05.05./07.05. Personen ohne Team müssen bis 07.05. ein Team finden.	08.05. Nur für Personen, die noch keinem Team zugewiesen sind.

Mehr Fragen!


Stellt eure Fragen
Über die Mailingliste
In der Vorlesung / Übung
Euren Tutor*innen
Mir (konitzny@ibr.cs.tu-bs.de)

Fragen?

Previously on AuD…

Problem vs Instanz

Problem	Instanz
Allgemeine Formulierung der Ein- und Ausgabe	Konkrete Werte für Ein- und Ausgabe
Eingabe: $z_1, \dots, z_n, Z, p_1, \dots, p_n$ Ausgabe: $S \subseteq \{1, \dots, n\}$ mit $\sum_{i \in S} z_i \leq Z$ und $\sum_{i \in S} p_i$ maximal.	Eingabe: $Z = 12$ und Objekte Ausgabe: $S = \{1,2,4,5\}$ denn $\sum_{i \in S} z_i = 9 \leq 12$ und $\sum_{i \in S} p_i = 11$
Lösung: Angabe eines Algorithmus	Lösung: Angabe konkreter Werte

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

Ist das bestmöglich?

Fractional Knapsack – Greedy Algorithmus

Eingabe: $z_i, ..., z_n, Z, p_i, ..., p_n$

Ausgabe: $x_i, ..., x_n \in [0, 1]$

mit

$$\sum_{i=1}^{n} z_i x_i \leq Z$$

und

$$\sum_{i=1}^{n} p_i x_i = Maximal$$

1: *Sortiere* $\{1,...,n\}$ *nach* $\frac{z_i}{p_i}$ *aufsteigend;*
Dies ergibt die Permutation $\pi(1), ..., \pi(n)$.
Setze $j = 1$.

2: ***while*** $(\sum_{i=1}^{j} z_{\pi(i)} \leq Z)$ ***do***

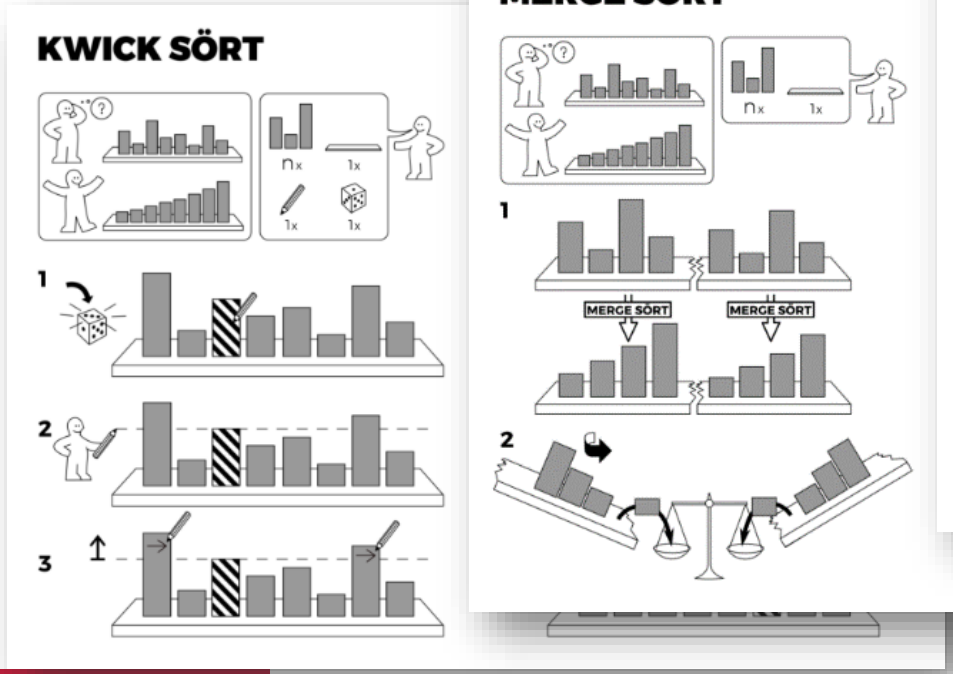
3: $\quad x_{\pi(j)} := 1$

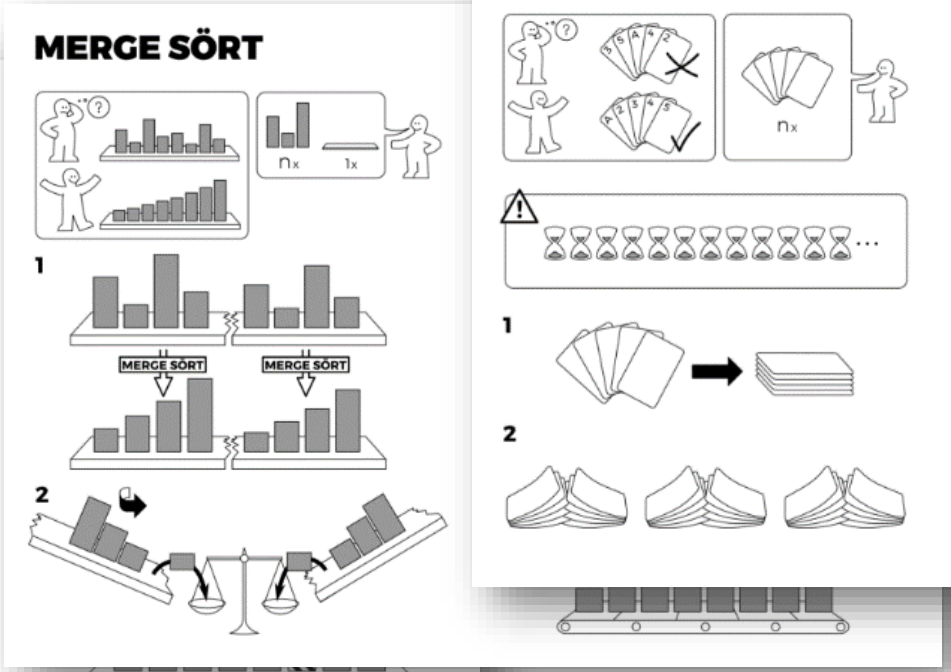
4: $\quad j := j + 1$

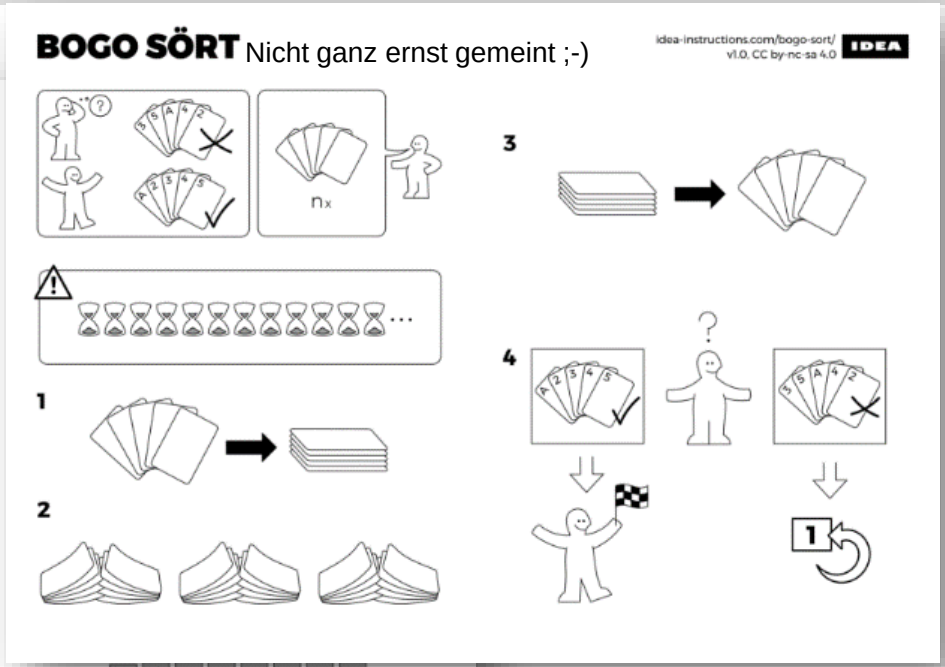
5: *Setze* $x_{\pi(j)} := \dfrac{Z - \sum_{i=1}^{j-1} z_{\pi(i)}}{z_{\pi(j)}}$

6: ***return***

Ein paar Sortierverfahren







Laufzeit – Sortieren

Wie lange dauern diese Algorithmen?

Algorithmus	Best-Case	Average-Case	Worst-Case
Quicksort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$
Mergesort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$

Theorem:
Jedes vergleichsbasierte Verfahren benötigt $\Omega(n \log n)$ Schritte.

Laufzeit

Sei $f(n)$ die Laufzeit eines Algorithmus mit Inputgröße n.

O-Notation:

Können wir garantieren, dass $f(n) \leq c_1 \cdot g(n)$ ab einem n_0 gilt, so schreiben wir $f(n) \in O\big(g(n)\big)$.

Maximale Laufzeit

Ω-Notation:

Können wir garantieren, dass $f(n) \geq c_2 \cdot g(n)$ ab einem n_0 gilt, so schreiben wir $f(n) \in \Omega\big(g(n)\big)$.

Mindestlaufzeit

Fractional Knapsack – Greedy Algorithmus

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

$Z = 12$ und

Eingabe: $z_i, ..., z_n, Z, p_i, ..., p_n$

Ausgabe: $x_i, ..., x_n \in [0, 1]$

mit

$$\sum_{i=1}^{n} z_i x_i \leq Z$$

und

$$\sum_{i=1}^{n} p_i x_i = Maximal$$

1: *Sortiere {1,...,n} nach* $\frac{z_i}{p_i}$ *aufsteigend;*
Dies ergibt die Permutation $\pi(1), ..., \pi(n)$.
Setze $j = 1$.

2: ***while*** $(\sum_{i=1}^{j} z_{\pi(i)} \leq Z)$ ***do***

3: $\quad x_{\pi(j)} := 1$

4: $\quad j := j + 1$

5: *Setze* $x_{\pi(j)} := \dfrac{Z - \sum_{i=1}^{j-1} z_{\pi(i)}}{z_{\pi(j)}}$

6: ***return***

Fractional Knapsack – Greedy Algorithmus

Eingabe: $z_i, ..., z_n, Z, p_i, ..., p_n$

Ausgabe: $x_i, ..., x_n \in [0, 1]$

mit

$$\sum_{i=1}^{n} z_i x_i \leq Z$$

und

$$\sum_{i=1}^{n} p_i x_i = Maximal$$

1: *Sortiere {1,...,n} nach* $\frac{z_i}{p_i}$ *aufsteigend;*
Dies ergibt die Permutation $\pi(1), ..., \pi(n)$.
Setze $j = 1$.
2: **while** $(\sum_{i=1}^{j} z_{\pi(i)} \leq Z)$ **do**
3: $x_{\pi(j)} := 1$
4: $j := j + 1$
5: *Setze* $x_{\pi(j)} := \frac{Z - \sum_{i=1}^{j-1} z_{\pi(i)}}{z_{\pi(j)}}$
6: **return**

$Z = 12$ und

i	1	2	3	4	5
z_i	2	1	7	4	2
p_i	3	1	4	5	2
$\frac{z_i}{p_i}$	$\frac{2}{3}$	1	$\frac{7}{4}$	$\frac{4}{5}$	1

Fractional Knapsack – Greedy Algorithmus

Eingabe: $z_i, ..., z_n, Z, p_i, ..., p_n$
Ausgabe: $x_i, ..., x_n \in [0, 1]$
mit

$$\sum_{i=1}^{n} z_i x_i \leq Z$$

und

$$\sum_{i=1}^{n} p_i x_i = Maximal$$

1: *Sortiere {1,...,n} nach* $\frac{z_i}{p_i}$ *aufsteigend;*
Dies ergibt die Permutation $\pi(1), ..., \pi(n)$.
Setze $j = 1$.
2: **while** $(\sum_{i=1}^{j} z_{\pi(i)} \leq Z)$ **do**
3: $x_{\pi(j)} := 1$
4: $j := j + 1$
5: *Setze* $x_{\pi(j)} := \frac{Z - \sum_{i=1}^{j-1} z_{\pi(i)}}{z_{\pi(j)}}$
6: **return**

$Z = 12$ und

i	1	2	3	4	5
z_i	2	1	7	4	2
p_i	3	1	4	5	2
$\frac{z_i}{p_i}$	$\frac{2}{3}$	1	$\frac{7}{4}$	$\frac{4}{5}$	1
RF	1	3	5	2	4

Fractional Knapsack – Greedy Algorithmus

Eingabe: $z_i, ..., z_n, Z, p_i, ..., p_n$
Ausgabe: $x_i, ..., x_n \in [0, 1]$
mit

$$\sum_{i=1}^{n} z_i x_i \leq Z$$

und

$$\sum_{i=1}^{n} p_i x_i = Maximal$$

1: *Sortiere {1,...,n} nach* $\frac{z_i}{p_i}$ *aufsteigend;*
Dies ergibt die Permutation $\pi(1), ..., \pi(n)$.
Setze $j = 1$.
2: **while** $(\sum_{i=1}^{j} z_{\pi(i)} \leq Z)$ **do**
3: $x_{\pi(j)} := 1$
4: $j := j + 1$
5: *Setze* $x_{\pi(j)} := \frac{Z - \sum_{i=1}^{j-1} z_{\pi(i)}}{z_{\pi(j)}}$
6: **return**

$Z = 12$ und

i	1	2	3	4	5
z_i	2	1	7	4	2
p_i	3	1	4	5	2
$\frac{z_i}{p_i}$	$\frac{2}{3}$	1	$\frac{7}{4}$	$\frac{4}{5}$	1
RF	1	3	5	2	4

j	$\pi(j)$	$x_{\pi(j)}$	$\sum_{}^{5} x_i z_i$	$Z - \sum_{}^{5} x_i z_i$	$\sum_{}^{5} x_i p_i$

Fractional Knapsack – Greedy Algorithmus

Eingabe: $z_i, ..., z_n, Z, p_i, ..., p_n$

Ausgabe: $x_i, ..., x_n \in [0, 1]$

mit

$$\sum_{i=1}^{n} z_i x_i \leq Z$$

und

$$\sum_{i=1}^{n} p_i x_i = Maximal$$

1: *Sortiere {1,...,n} nach* $\frac{z_i}{p_i}$ *aufsteigend;*
Dies ergibt die Permutation $\pi(1), ..., \pi(n)$.
Setze $j = 1$.
2: **while** $(\sum_{i=1}^{j} z_{\pi(i)} \leq Z)$ **do**
3: $x_{\pi(j)} := 1$
4: $j := j + 1$
5: *Setze* $x_{\pi(j)} := \frac{Z - \sum_{i=1}^{j-1} z_{\pi(i)}}{z_{\pi(j)}}$
6: **return**

$Z = 12$ und

i	1	2	3	4	5
z_i	2	1	7	4	2
p_i	3	1	4	5	2
$\frac{z_i}{p_i}$	$\frac{2}{3}$	1	$\frac{7}{4}$	$\frac{4}{5}$	1
RF	1	3	5	2	4

j	$\pi(j)$	$x_{\pi(j)}$	$\sum_{i=1}^{5} x_i z_i$	$Z - \sum_{i=1}^{5} x_i z_i$	$\sum_{i=1}^{5} x_i p_i$
1	1	1	2	10	3

Fractional Knapsack – Greedy Algorithmus

Eingabe: $z_i, ..., z_n, Z, p_i, ..., p_n$
Ausgabe: $x_i, ..., x_n \in [0, 1]$
mit

$$\sum_{i=1}^{n} z_i x_i \leq Z$$

und

$$\sum_{i=1}^{n} p_i x_i = Maximal$$

1: *Sortiere {1,...,n} nach* $\frac{z_i}{p_i}$ *aufsteigend;*
Dies ergibt die Permutation $\pi(1), ..., \pi(n)$.
Setze $j = 1$.
2: **while** $(\sum_{i=1}^{j} z_{\pi(i)} \leq Z)$ **do**
3: $x_{\pi(j)} := 1$
4: $j := j + 1$
5: *Setze* $x_{\pi(j)} := \frac{Z - \sum_{i=1}^{j-1} z_{\pi(i)}}{z_{\pi(j)}}$
6: **return**

$Z = 12$ und

i	1	2	3	4	5
z_i	2	1	7	4	2
p_i	3	1	4	5	2
$\frac{z_i}{p_i}$	$\frac{2}{3}$	1	$\frac{7}{4}$	$\frac{4}{5}$	1
RF	1	3	5	2	4

j	$\pi(j)$	$x_{\pi(j)}$	$\sum_{i=1}^{5} x_i z_i$	$Z - \sum_{i=1}^{5} x_i z_i$	$\sum_{i=1}^{5} x_i p_i$
1	1	1	2	10	3
2	4	1	6	6	8

Fractional Knapsack – Greedy Algorithmus

Eingabe: $z_i, ..., z_n, Z, p_i, ..., p_n$
Ausgabe: $x_i, ..., x_n \in [0, 1]$
mit

$$\sum_{i=1}^{n} z_i x_i \leq Z$$

und

$$\sum_{i=1}^{n} p_i x_i = Maximal$$

1: *Sortiere {1,...,n} nach* $\frac{z_i}{p_i}$ *aufsteigend;*
Dies ergibt die Permutation $\pi(1), ..., \pi(n)$.
Setze $j = 1$.
2: **while** $(\sum_{i=1}^{j} z_{\pi(i)} \leq Z)$ **do**
3: $x_{\pi(j)} := 1$
4: $j := j + 1$
5: *Setze* $x_{\pi(j)} := \frac{Z - \sum_{i=1}^{j-1} z_{\pi(i)}}{z_{\pi(j)}}$
6: **return**

$Z = 12$ und

i	1	2	3	4	5
z_i	2	1	7	4	2
p_i	3	1	4	5	2
$\frac{z_i}{p_i}$	$\frac{2}{3}$	1	$\frac{7}{4}$	$\frac{4}{5}$	1
RF	1	3	5	2	4

j	$\pi(j)$	$x_{\pi(j)}$	$\sum_{i=1}^{5} x_i z_i$	$Z - \sum_{i=1}^{5} x_i z_i$	$\sum_{i=1}^{5} x_i p_i$
1	1	1	2	10	3
2	4	1	6	6	8
3	2	1	7	5	9

Fractional Knapsack – Greedy Algorithmus

i	1	2	3	4	5
z_i	2	1	7	4	2
p_i	3	1	4	5	2
$\frac{z_i}{p_i}$	$\frac{2}{3}$	1	$\frac{7}{4}$	$\frac{4}{5}$	1
RF	1	3	5	2	4

$Z = 12$ und

Eingabe: $z_i, ..., z_n, Z, p_i, ..., p_n$

Ausgabe: $x_i, ..., x_n \in [0, 1]$

mit

$$\sum_{i=1}^{n} z_i x_i \leq Z$$

und

$$\sum_{i=1}^{n} p_i x_i = Maximal$$

1: *Sortiere {1,...,n} nach* $\frac{z_i}{p_i}$ *aufsteigend;*
Dies ergibt die Permutation $\pi(1), ..., \pi(n)$.
Setze $j = 1$.
2: ***while*** $(\sum_{i=1}^{j} z_{\pi(i)} \leq Z)$ ***do***
3: $x_{\pi(j)} := 1$
4: $j := j + 1$
5: *Setze* $x_{\pi(j)} := \frac{Z - \sum_{i=1}^{j-1} z_{\pi(i)}}{z_{\pi(j)}}$
6: ***return***

j	$\pi(j)$	$x_{\pi(j)}$	$\sum_{i=1}^{5} x_i z_i$	$Z - \sum_{i=1}^{5} x_i z_i$	$\sum_{i=1}^{5} x_i p_i$
1	1	1	2	10	3
2	4	1	6	6	8
3	2	1	7	5	9
4	5	1	9	3	11

Fractional Knapsack – Greedy Algorithmus

i	1	2	3	4	5
z_i	2	1	7	4	2
p_i	3	1	4	5	2
$\frac{z_i}{p_i}$	$\frac{2}{3}$	1	$\frac{7}{4}$	$\frac{4}{5}$	1
RF	1	3	5	2	4

$Z = 12$ und

Eingabe: $z_i, ..., z_n, Z, p_i, ..., p_n$

Ausgabe: $x_i, ..., x_n \in [0, 1]$

mit

$$\sum_{i=1}^{n} z_i x_i \leq Z$$

und

$$\sum_{i=1}^{n} p_i x_i = Maximal$$

1: *Sortiere {1,...,n} nach* $\frac{z_i}{p_i}$ *aufsteigend;*
Dies ergibt die Permutation $\pi(1), ..., \pi(n)$.
Setze $j = 1$.
2: ***while*** $(\sum_{i=1}^{j} z_{\pi(i)} \leq Z)$ ***do***
3: $x_{\pi(j)} := 1$
4: $j := j + 1$
5: *Setze* $x_{\pi(j)} := \frac{Z - \sum_{i=1}^{j-1} z_{\pi(i)}}{z_{\pi(j)}}$
6: ***return***

j	$\pi(j)$	$x_{\pi(j)}$	$\sum_{i=1}^{5} x_i z_i$	$Z - \sum_{i=1}^{5} x_i z_i$	$\sum_{i=1}^{5} x_i p_i$
1	1	1	2	10	3
2	4	1	6	6	8
3	2	1	7	5	9
4	5	1	9	3	11
5	3	3/7	12	0	$12 + 5/7$

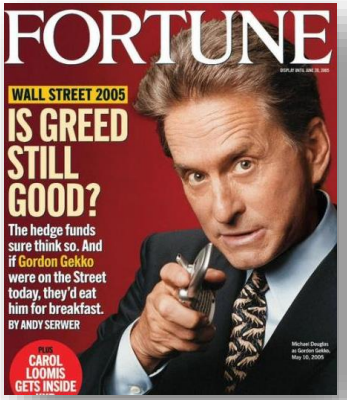
Knapsack – Greedy Algorithmen

Greedy für fraktionales Knapsack

https://c1.staticflickr.com/9/8246/8491125952_c2c7c854d8.jpg

Greedy ist optimal!

Greedy für ganzzahliges Knapsack



http://coverjunkie.com/uploads/1315147575.jpg

Ist Greedy noch optimal?