



Technische  
Universität  
Braunschweig



# Algorithmen und Datenstrukturen 2 – Übung #0

Arne Schmidt  
29.04.2020

# Übersicht

Organisation

Wiederholung AuD 1

Greedy für Fractional Knapsack

# Organisation

Homepage:

<https://www.ibr.cs.tu-bs.de/courses/ss20/aud2/>

Dort gibt es:

Aktuelle Informationen,  
Hausaufgaben,  
Notizen zu Übungen  
und Vorlesung, ...

Es gibt ein **Skript**.

Fehler im Skript per Mail an mich melden.

The screenshot shows the course page for 'Algorithmen und Datenstrukturen II' in the summer semester 2020. The page includes a navigation menu with links for 'Impressum', 'Datenschutz', 'English', 'aschmidt', 'Schnellsuche', and 'Edit'. A sidebar on the left lists various categories like 'News', 'Wir über uns', 'Connected and Mobile Systems', 'Algorithmen', 'Lehrveranstaltungen', 'Abschlussarbeiten', 'Projekte', 'Veröffentlichungen', 'Verteilte Systeme', 'Mikroprozessorlabor', 'Studium', 'Sommersemester 2020', 'Wintersemester 2019/2020', 'Abschlussarbeiten', 'Service', 'Spin-Offs', and 'Forschungsverbände'. The main content area displays course details: Semester (Sommersemester 2020), Modulnummer (INF-ALG-23), Veranstaltungsnr. (INF-ALG-042, INF-ALG-043, INF-ALG-044), Studiengänge (Bachelor Wirtschaftsinformatik, Bachelor Informations-Systemtechnik, Bachelor Informatik), IBR Gruppe (ALG (Prof. Fekete)), and Art (Vorlesung/Übung). The dozent is Prof. Dr. Sándor P. Fekete, and the assistent is Arne Schmidt. The page also lists the number of LPs (5), SWS (2+1+1), and the schedule for the course. A table shows the course groups and their respective times and rooms. The page concludes with information on the start date, prerequisites, language, and the grading system.


www.ibr.cs.tu-bs.de/courses/ss20/aud2/index.xml


Impressum Datenschutz English aschmidt Schnellsuche Edit

News  
Wir über uns  
Connected and Mobile Systems  
Algorithmen  
Lehrveranstaltungen  
Abschlussarbeiten  
Projekte  
Veröffentlichungen  
Verteilte Systeme  
Mikroprozessorlabor  
Studium  
Sommersemester 2020  
Wintersemester 2019/2020  
Abschlussarbeiten  
Service  
Spin-Offs  
Forschungsverbände

### Algorithmen und Datenstrukturen II

Semester : Sommersemester 2020  
Modulnummer : INF-ALG-23  
Veranstaltungsnr. : INF-ALG-042, INF-ALG-043, INF-ALG-044  
Studiengänge : Bachelor Wirtschaftsinformatik, Bachelor Informations-Systemtechnik, Bachelor Informatik  
IBR Gruppe : ALG (Prof. Fekete)  
Art : Vorlesung/Übung

Dozent :  Prof. Dr. Sándor P. Fekete  
Abteilungsleiter  
s.fekete@tu-bs.de  
+49 531 3913111  
Raum 335

Assistent :  Arne Schmidt  
Wissenschaftlicher Mitarbeiter  
aschmidt@ibr.cs.tu-bs.de  
+49 531 3913115  
Raum 319

LP : 5  
SWS : 2+1+1

Ort & Zeit : Vorlesung: Donnerstags, 15.00 Uhr, Raum: <https://bbb.ibr.cs.tu-bs.de/b/pro-aj2-agg>  
Große Übung: Mittwochs, 15.00 Uhr, Raum: TBA  
Kleine Übung: TBA

Gruppe	Termin (14-tägig)	Raum	Tutor
1	Mittwoch, 09.45 - 11.15	TBA	Tobias Wallner
2	Mittwoch, 13.15 - 14.45	TBA	Dennis Luck
3	Freitag, 11.30 - 13.00	TBA	Maurice Semren
4	Freitag, 13.15 - 14.45	TBA	Ahmad Al-Mekhlafy

Beginn : Erste Vorlesung: 23.04.20  
Erste große Übung: 29.04.20  
Erste kleine Übung: TBA

Voraussetzungen : Keine

Sprache : Deutsch

Scheinerwerb : Studienleistung: Mindestens 50 Prozent der Hausaufgabenpunkte  
Prüfungsleistung: Klausur am Ende des Semesters

Anmeldung : Die Anmeldung zu den Übungsgruppen ist hier bis 29.04.2020 um 23:59 möglich. Erhalten wir mehrere

# 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 **Arne Schmidt**.

**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

# Anmeldung

Anmeldung zu kleinen Übungen auf der Homepage

Gruppe	Termin (14-tägig)	Raum	Tutor
1	Mittwoch, 09:45 - 11:15	TBA	Tobias Wallner
2	Mittwoch, 13:15 - 14:45	TBA	Dennis Luck
3	Freitag, 11:30- 13:00	TBA	Maurice Semren
4	Freitag, 13:15 - 14:45	TBA	Ahmad Al-Mekhlafy

Läuft bis 29.04.20.  
Einteilung erfolgt voraussichtlich morgen.

# Anmeldung

Die Gruppenzuweisung wird an jede Person einzeln per eMail versandt (voraussichtlich am 30.04.2020). Stellt also sicher, dass eure eMail-Adresse korrekt ist. Zur Sicherheit versenden wir eine Mail an die angegebene Adresse und zusätzlich eine an die tu-bs-Adresse über eure y-Nummer.

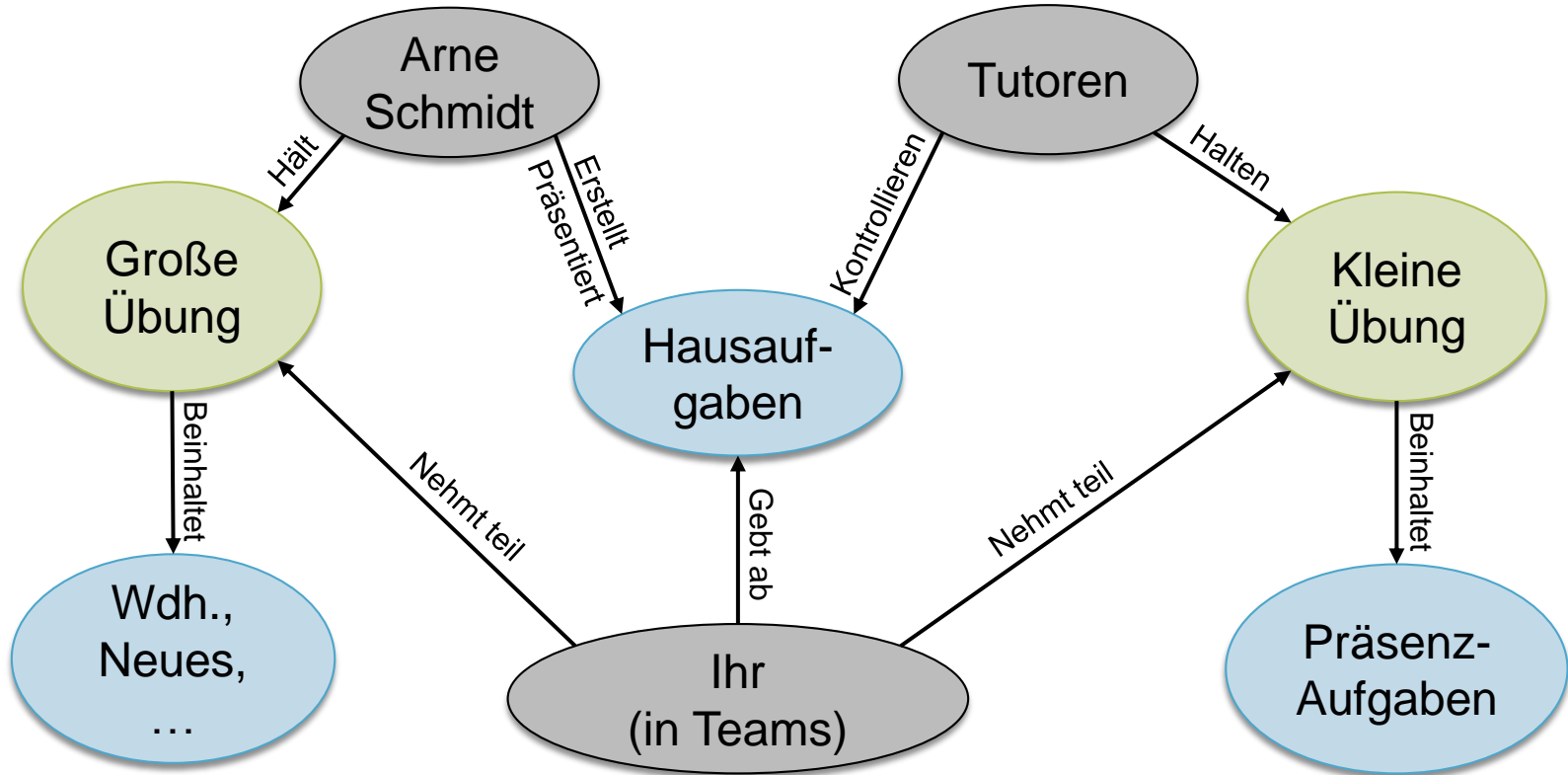
Weitere Informationen zur Anmeldung:

- Es wird Hausaufgaben geben, die in Gruppen mit voraussichtlich 3 bis 5 Personen bearbeitet und abgegeben werden.
- Es können bis zu 4 Wunschartner angegeben werden. Eine Gruppe wird dann transitiv ermittelt, d.h., möchte Person A mit Person B in eine Gruppe und B mit Person C, dann kommen A, B und C in eine Gruppe. Sollten dabei Gruppen mit mindestens 6 Personen entstehen werden diese per Zufall aufgeteilt. Zu kleine Gruppen werden so weit es geht aufgefüllt.
- Jeder der Wunschartner muss sich separat anmelden.
- Sollte eine Anmeldung nicht erfolgreich sein (roter Text erscheint unter dem Anmeldeformular), betrachten Sie folgende Hinweise:
  - Die Matrikelnummer ist eine siebenstellige Zahl. Es sind nicht die Zahlen der y-Nummer.
  - Die y-Nummer ist ein Wort bestehend aus 'y' und sieben folgenden Zahlen.
  - Leerzeichen sind im Mail-, Matrikel-, Semester- und yNummerfeld nicht erlaubt.
  - Sollten weiterhin Fehler auftreten, senden Sie eine Mail an [Arne Schmidt](#)

Nachname *	<input type="text"/>
Vorname *	<input type="text"/>
Matrikelnummer *	<input type="text"/>
Studiengang *	Informatik
Semester *	<input type="text"/>
y-Nummer *	<input type="text"/>
E-Mail *	<input type="text"/>
Mittwoch, 09:45	<input type="checkbox"/>
Mittwoch, 13:15	<input type="checkbox"/>
Freitag, 11:30	<input type="checkbox"/>
Freitag, 13:15	<input type="checkbox"/>
y-Nummer Gruppenmitglied 2 :	<input type="text"/>
y-Nummer Gruppenmitglied 3 :	<input type="text"/>
y-Nummer Gruppenmitglied 4 :	<input type="text"/>
y-Nummer Gruppenmitglied 5 :	<input type="text"/>

Mit Absenden dieses Formular stimmen Sie der Verarbeitung der von Ihnen selbst angegebenen Daten nur zum Zwecke der hier erfolgten Erhebung zu. Außerdem nehmen Sie die [Datenschutzerklärung der TU Braunschweig](#) und die [Datenschutzerklärung des IBR](#) zur Kenntnis nach denen die Verarbeitung Ihrer Daten erfolgt.

# Übungsbetrieb



# Hausaufgaben I

Voraussichtlich 6 Blätter

- 1 unbewertet
- 5 bewertet

30 Punkte pro bewertetem Blatt

Insgesamt 150 Punkte erreichbar.

50% müssen erreicht werden.

Keine Mindestpunktzahl pro Blatt!

Abgabefrist zählt das Datum auf dem jeweiligen Blatt.

Abgabe als Team per Mail an die Tutoren.  
(<https://www.ibr.cs.tu-bs.de/alg/index.html>)



# Hausaufgaben II

Die Ausarbeitungen müssen in Teams (3-5 Personen) abgegeben werden.

## Vorschlag von uns:

1. Selbst Lösungen erarbeiten
2. Lösungen in Gruppen diskutieren
3. Gesamtlösung erarbeiten und abschicken



**Wichtig:**  
Kommunikation  
untereinander!

Zu spät eingereichte Abgaben werden mit 0 Punkten bewertet

# 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)

## 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

Bis heute 23:59

Einteilung  
in Gruppen

Voraussichtlich  
morgen

0. kl.  
Übung

13.05./15.05.

Personen ohne  
Team müssen bis  
17.05. ein Team  
finden oder mit  
anderen gründen.

Einteilung  
in Teams

18.05.

Nur für Personen,  
die noch keinem  
Team zugewiesen  
sind.

# Mehr Fragen!

Stellt eure Fragen:

Über die Mailingliste

In: Vorlesung/Übung

Euren Tutoren

Per Mail an [aschmidt@ibr.cs.tu-bs.de](mailto:aschmidt@ibr.cs.tu-bs.de)

Was ist mit  
der Klausur?

**Fragen?**



# Previously on AuD...

# Problem vs Instanz

## Problem

Allgemeine Formulierung der 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.

Lösung: Angabe eines Algorithmus

## Instanz

Konkrete Werte für Ein- und Ausgabe

Eingabe:  $Z = 12$  und Objekte

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

Ausgabe:

$S = \{1, 2, 4, 5\}$  denn  
 $\sum_{i \in S} z_i = 9 \leq 12$  und  
 $\sum_{i \in S} p_i = 11$

Ist das bestmöglich?

Lösung: Angabe konkreter Werte

# Fractional Knapsack – Greedy Algorithmus

---

**Eingabe:**  $z_1, \dots, z_n, Z, p_1, \dots, p_n$

**Ausgabe:**  $x_1, \dots, x_n \in [0, 1]$

mit

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

und

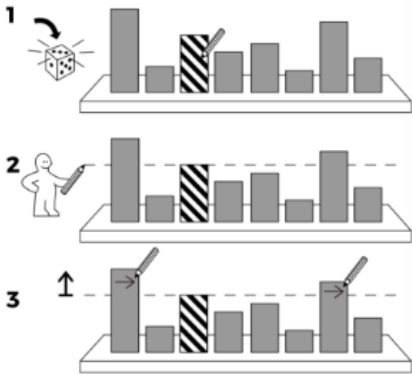
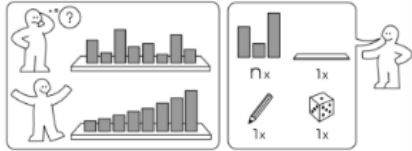
$$\sum_{i=1}^n p_i x_i = \text{Maximal}$$

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

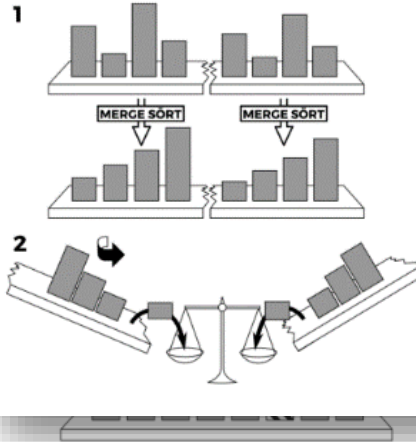
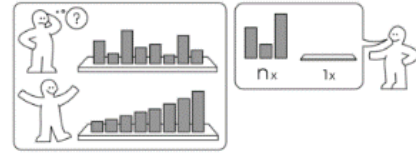


# Ein paar Sortierverfahren

## KWICK SÖRT

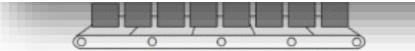
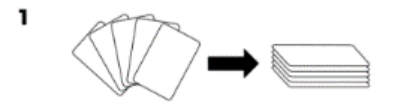
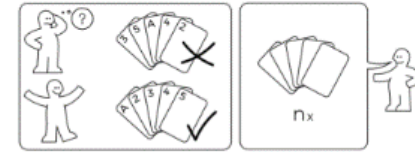


## MERGE SÖRT



## BOGO SÖRT Nicht ganz ernst gemeint ;-)

idea-instructions.com/bogo-sort/  
v1.0. CC BY-NC-SA 4.0 **IDEA**



# 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

$$f(n) \in O(g(n)).$$

Maximale Laufzeit

## **$\Omega$ -Notation:**

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

$$f(n) \in \Omega(g(n)).$$

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

**Eingabe:**  $z_1, \dots, z_n, Z, p_1, \dots, p_n$

$Z = 12$  und

**Ausgabe:**  $x_1, \dots, x_n \in [0, 1]$

mit

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

und

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

1: Sortiere  $\{1, \dots, n\}$  nach  $\frac{z_i}{p_i}$  aufsteigend;

Dies ergibt die Permutation  $\pi(1), \dots, \pi(n)$ .

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**

# Fractional Knapsack – Greedy Algorithmus

**Eingabe:**  $z_1, \dots, z_n, Z, p_1, \dots, p_n$

**Ausgabe:**  $x_1, \dots, x_n \in [0, 1]$

mit

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

und

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

1: Sortiere  $\{1, \dots, n\}$  nach  $\frac{z_i}{p_i}$  aufsteigend;

Dies ergibt die Permutation  $\pi(1), \dots, \pi(n)$ .

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:**  $z_1, \dots, z_n, Z, p_1, \dots, p_n$

**Ausgabe:**  $x_1, \dots, x_n \in [0, 1]$

mit

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

und

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

1: Sortiere  $\{1, \dots, n\}$  nach  $\frac{z_i}{p_i}$  aufsteigend;

Dies ergibt die Permutation  $\pi(1), \dots, \pi(n)$ .

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:**  $z_1, \dots, z_n, Z, p_1, \dots, p_n$

**Ausgabe:**  $x_1, \dots, x_n \in [0, 1]$

mit

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

und

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

1: Sortiere  $\{1, \dots, n\}$  nach  $\frac{z_i}{p_i}$  aufsteigend;

Dies ergibt die Permutation  $\pi(1), \dots, \pi(n)$ .

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

$j$	$\pi(j)$	$x_{\pi(j)}$	$\sum_{i=1}^j x_i z_i$	$Z - \sum_{i=1}^j x_i z_i$	$\sum_{i=1}^j x_i p_i$
1	2	1	1	11	1
2	1	1	3	9	4
3	3	0.5	5	7	7
4	4	0	5	7	7
5	5	0	5	7	7

# Fractional Knapsack – Greedy Algorithmus

**Eingabe:**  $z_1, \dots, z_n, Z, p_1, \dots, p_n$

**Ausgabe:**  $x_1, \dots, x_n \in [0, 1]$

mit

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

und

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

1: Sortiere  $\{1, \dots, n\}$  nach  $\frac{z_i}{p_i}$  aufsteigend;

Dies ergibt die Permutation  $\pi(1), \dots, \pi(n)$ .

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:**  $z_1, \dots, z_n, Z, p_1, \dots, p_n$

**Ausgabe:**  $x_1, \dots, x_n \in [0, 1]$

mit

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

und

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

1: Sortiere  $\{1, \dots, n\}$  nach  $\frac{z_i}{p_i}$  aufsteigend;

Dies ergibt die Permutation  $\pi(1), \dots, \pi(n)$ .

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:**  $z_1, \dots, z_n, Z, p_1, \dots, p_n$

**Ausgabe:**  $x_1, \dots, x_n \in [0, 1]$

mit

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

und

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

1: Sortiere  $\{1, \dots, n\}$  nach  $\frac{z_i}{p_i}$  aufsteigend;

Dies ergibt die Permutation  $\pi(1), \dots, \pi(n)$ .

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

**Eingabe:**  $z_1, \dots, z_n, Z, p_1, \dots, p_n$

**Ausgabe:**  $x_1, \dots, x_n \in [0, 1]$

mit

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

und

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

1: Sortiere  $\{1, \dots, n\}$  nach  $\frac{z_i}{p_i}$  aufsteigend;

Dies ergibt die Permutation  $\pi(1), \dots, \pi(n)$ .

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
4	5	1	9	3	11

# Fractional Knapsack – Greedy Algorithmus

**Eingabe:**  $z_1, \dots, z_n, Z, p_1, \dots, p_n$

**Ausgabe:**  $x_1, \dots, x_n \in [0, 1]$

mit

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

und

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

1: Sortiere  $\{1, \dots, n\}$  nach  $\frac{z_i}{p_i}$  aufsteigend;

Dies ergibt die Permutation  $\pi(1), \dots, \pi(n)$ .

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
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](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?