



Technische
Universität
Braunschweig



Algorithmen und Datenstrukturen 2 – Übung #0

Matthias Konitzny

27.04.2022

Homepage

<https://www.ibr.cs.tu-bs.de/courses/ss22/aud2/>

The screenshot shows the course page for INF-ALG-23. It includes the following information:

- Semester:** Sommersemester 2022
- Modulnummer:** INF-ALG-23
- Veranstaltungsnummer:** 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, Abteilungleiter, s.fekete@tu-bs.de, +49 531 3913111, Raum 335
- Assistent:** Matthias Konitzny, Wissenschaftlicher Mitarbeiter, konitzny@ibr.cs.tu-bs.de, +49 531 3913113, Raum 333
- LP:** 5
- SWS:** 2+1+1
- Ort & Zeit:** Vorlesung: Dienstags, 9:45 Uhr, Raum: PK 15.1 (bzw. UP 3.007); Große Übung: Mittwochs, 15:00 Uhr, Raum: PK 11.2; Kleine Übung: TBA
- Beginn:** Erste Vorlesung: 26.04.22

On the right side of the page, there is a navigation menu with the following items:

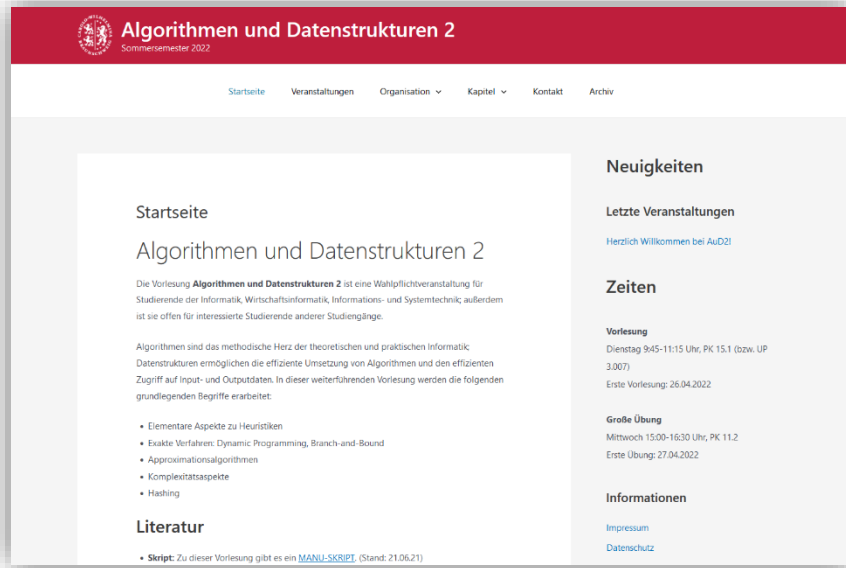
- ^ Institut für Betriebssysteme und Rechnerverbund
- News
- ^ Wir über uns
- ^ Connected and Mobile Systems
- ^ Verteilte Systeme
- ^ Algorithmik
- Mikroprozessorlabor
- ^ Studium
 - Sommersemester 2022
 - Wintersemester 2021/2022
 - Abschlussarbeiten
- ^ Service
- Spin-Offs
- ^ Forschungsverbände

Dort findet ihr

- Allgemeine Informationen
- Aktuelle Hinweise

Materialseite

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



The screenshot shows the website for 'Algorithmen und Datenstrukturen 2' in the summer semester of 2022. The page has a red header with the course title and a navigation menu. The main content area is divided into sections: 'Startseite' with a title and introductory text, 'Neuigkeiten' with 'Letzte Veranstaltungen' and a welcome message, 'Zeiten' with 'Vorlesung' and 'Große Übung' details, and 'Literatur' with a note about a script. The text is in German and provides details about the course structure and schedule.

Dort findet ihr

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

Kommunikation

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 plaintext.

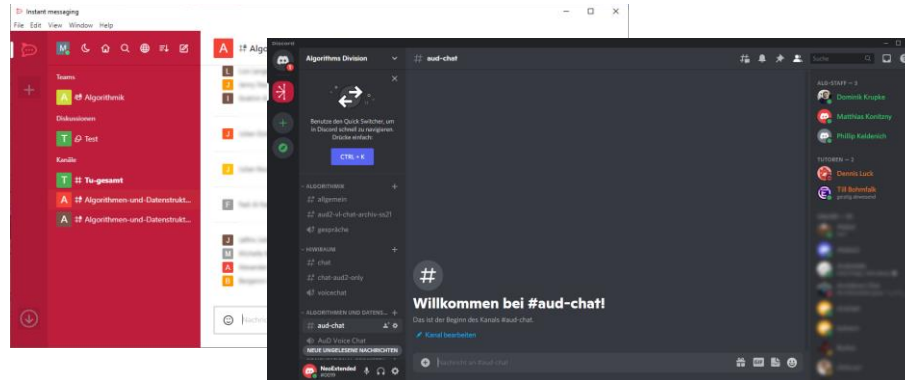
If you choose not to enter a password, one will be automatically generated for you, and it will be sent to you once you've confirmed your subscription. You can always request a mail-back of your password when you edit your personal options.

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



Mailingliste

- Anmeldung über Homepage
- Kurzfristige Information zur Vorlesung/Übung/Klausur/etc.
- Bietet Möglichkeit Fragen zu stellen

Messenger

- Fragen und Austausch in Echtzeit



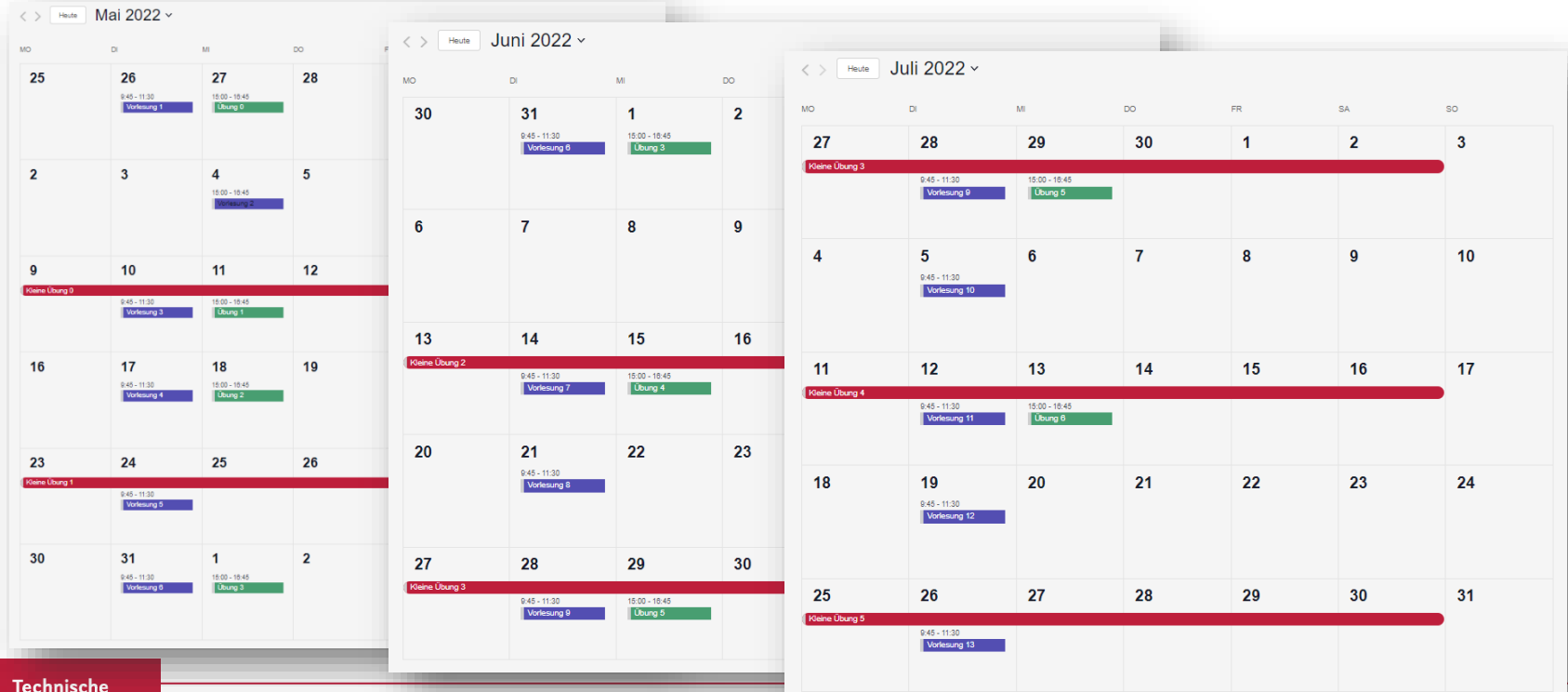
[Discord](https://discord.com)



[TU-Messenger](#)

Semesterkalender

<https://aud.ibr.cs.tu-bs.de/events/>



Anmeldung

Anmeldung zu kleinen Übungen auf der Homepage

Gruppe	Termin (14-täglich)	Raum	Tutor
01	Mittwoch: 09:45 – 11:15	IZ 305	TBD
02	Mittwoch: 13:15 – 14:45	IZ 305	TBD
03	Freitag: 11:30 – 13:00	IZ 305	TBD
04	Freitag: 13:15 – 14:45	IZ 305	TBD

Anmeldung

Anmeldung Kleine Übungen

Allgemein / Von konitzny

Die Anmeldung für die kleinen Übungen ist ab jetzt bis zum 08.05.22 geöffnet. Anschließend erhalten alle Teilnehmenden am 09.05.22 eine EMail mit der zugeteilten Übung und Gruppe.

Weitere Informationen

- Die Hausaufgaben sollen in 2-Personen-Gruppen erledigt werden.
- Die Wunschpartner*innen werden nur zugeteilt, wenn beide Partner*innen die Daten des jeweils anderen **korrekt** angeben. Ansonsten wird eine zufällige Person automatisch ausgewählt.
- Bitte gebt als Email-Adresse eure TU-BS Adresse an.
- Bitte achtet darauf, dass auch eure eigenen 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](#)

Nachname

Vorname

Matrikelnummer

Studiengang
Informatik

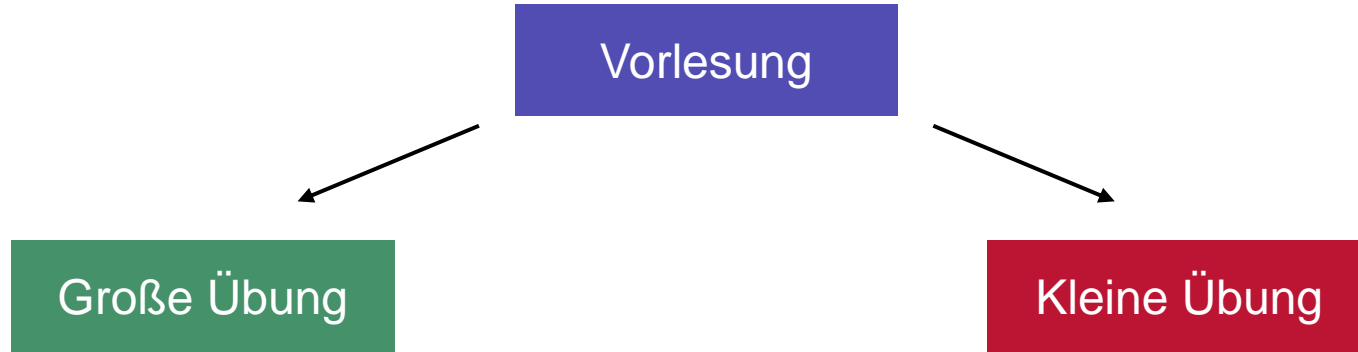
Semester
1

y-Nummer

E-Mail-Adresse

Falls die Studienleistung bereits bestanden wurde, können Hausaufgaben alleine abgegeben werden.
 Ich habe die Studienleistung bereits bestanden

Termine
 Mittwoch 9:45 Uhr
 Mittwoch 13:15 Uhr
 Freitag 11:30 Uhr
 Freitag 13:15 Uhr



- Aufarbeitung der Inhalte
- Zusätzliche Inhalte
- Beantwortung von Fragen
- Interaktion!

- Vertiefung der Inhalte
- Selbständiges Arbeiten
- Besprechung von Hausaufgaben

Hausaufgaben und Übungsblätter

5 verpflichtende Hausaufgabenblätter

- Studienleistung
- 20 Punkte pro Blatt
- Bearbeitung in 2er Gruppen
- Studienleistung: 50% der Gesamtpunkte

6 freiwillige Übungsblätter

- Zusätzliche Vertiefung
- Prüfungsvorbereitung

Algorithmen und Datenstrukturen 2
Prof. Dr. Sándor Péter
Matthias Konitzny

SoSe 2022
Abgabe: 17.05.22
Rückgabe: ab 25.05.22

Hausaufgabenblatt 1

Abgabe der Lösungen bis zum 17.05.22 um 16:00 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 (FRACTIONAL KNAPSACK): (5+4 Punkte)
In dieser Aufgabe betrachten wir FRACTIONAL KNAPSACK.

a) Sei $Z = 30$, und seien die sieben Objekte mit folgenden Werten gegeben:

i	1	2	3	4	5	6	7
w_i	6	7	10	4	8	5	9
p_i	2	7	11	7	6	4	5

- Studienleistung ist **keine** Voraussetzung, um an der Prüfung teilzunehmen.
- Studienleistung ist **eine** Voraussetzung, um das Modul abzuschließen.
- Studienleistung ist nicht benotet und fließt nicht in die Prüfung ein.

Hausaufgaben abgeben

Wozu Hausaufgaben?

Die Hausaufgaben dienen Euch (nicht uns) zur Vorbereitung auf die Klausur.

- Ideale Nachbereitung der Vorlesungsinhalte
- Zeitersparnis bei der Prüfungsvorbereitung
- Direktes Feedback, über euren aktuellen Lernstand.

- Zu späte Abgaben: 0 Punkte
- Falscher Abgabeschrank: 0 Punkte
- Mit Bleistift oder rot geschriebene Teile werden nicht gewertet
- Eine Abgabe pro Gruppe
- In Ausnahmefällen: Abgaben per Mail

Hausaufgaben



Hausaufgabenschrank

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

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

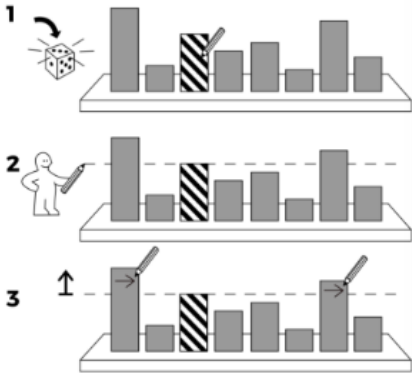
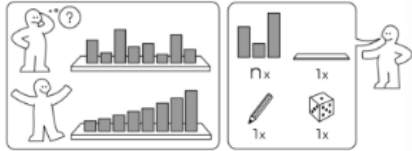
und

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

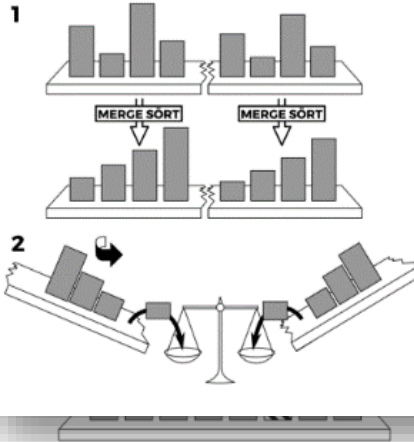
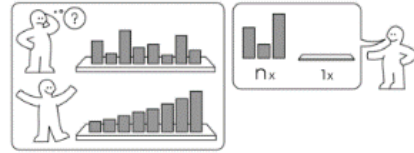
- 1: **Sortiere** $\{1, \dots, n\}$ nach $\frac{z_i}{p_i}$ aufsteigend;
Dies ergibt die Permutation $\pi(1), \dots, \pi(n)$.
Setze $j = 1$.
 - 2: **while** $(\sum_{i=1}^j z_{\pi(i)} \leq Z)$ **do**
 - 3: $x_{\pi(j)} := 1$
 - 4: $j := j + 1$
 - 5: Setze $x_{\pi(j)} := \frac{Z - \sum_{i=1}^{j-1} z_{\pi(i)}}{z_{\pi(j)}}$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

Ein paar Sortierverfahren

KWICK SÖRT

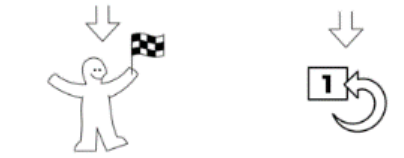
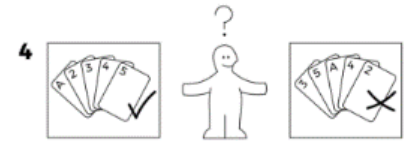
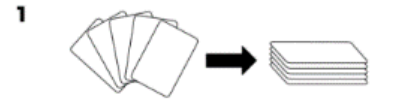
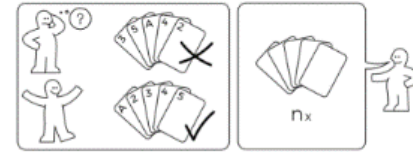


MERGE SÖRT



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 .

Maximale Laufzeit

O -Notation

Können wir garantieren, dass

$$f(n) \leq c_1 \cdot g(n)$$

ab einem n_0 gilt, so schreiben wir

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

Mindestlaufzeit

Ω -Notation

Können wir garantieren, dass

$$f(n) \geq c_2 \cdot g(n)$$

ab einem n_0 gilt, so schreiben wir

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

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

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