



Technische
Universität
Braunschweig



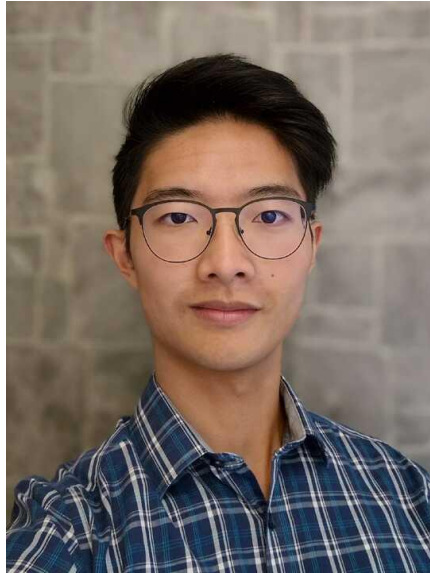
Algorithmen und Datenstrukturen 2 – Übung #0

Ramin Kosfeld und Chek-Manh Loi

15.04.2026

Wir sind ...

Chek-Manh



loi@ibr.cs.tu-bs.de

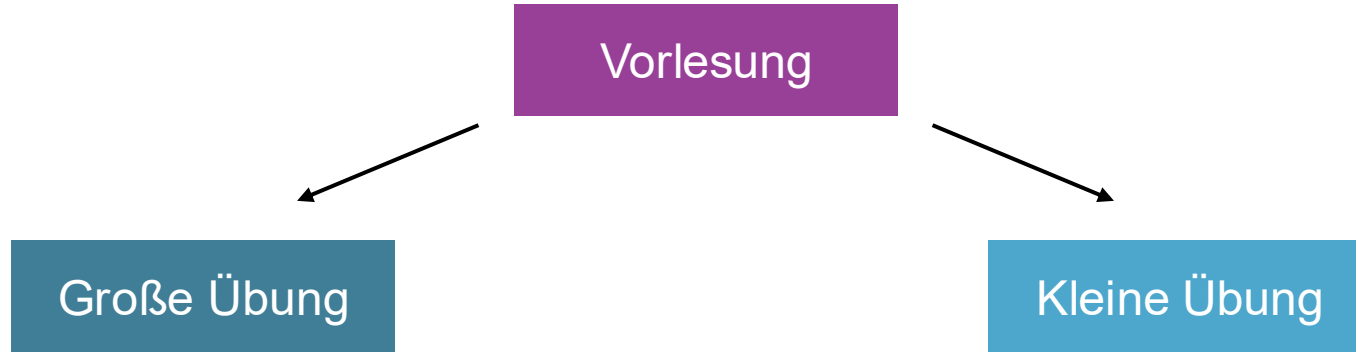
Ramin



kosfeld@ibr.cs.tu-bs.de

Organisation

Formate



Tafelübungen im Hörsaal (mit allen)

- Ca. alle 2 Wochen
- Wiederholung und Vertiefung der Inhalte, Beispiele, Ergänzungen
- Fragen klären

Kleingruppen (Seminarräume)

- Alle 2 Wochen
- Betreut von Übungsleiter*in
- Besprechung von Hausaufgaben
- Fragen klären und Inhalte üben

Hausaufgaben und Präsenzblätter

5 verpflichtende Hausaufgabenblätter

- Alle 2 Wochen
- Einzelabgaben
- 20 Punkte pro Blatt
- Korrigiert zurück :)
- Studienleistung

6 freiwillige Präsenzblätter

- Zusätzliche Vertiefung

- Studienleistung: 50% der Gesamtpunkte (Also: Ab insgesamt 50 Punkten bestanden)
 - 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.

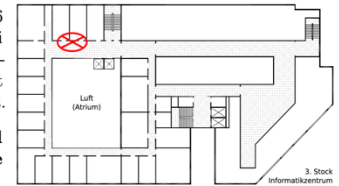
Algorithmen und Datenstrukturen 2
Prof. Sándor Fekete
Ramin Kosfeld
Chek-Manh Loi

Sommer 2026
Abgabe: 08.05.2026
Rückgabe: ab 11.05.2026

Hausaufgabenblatt 1

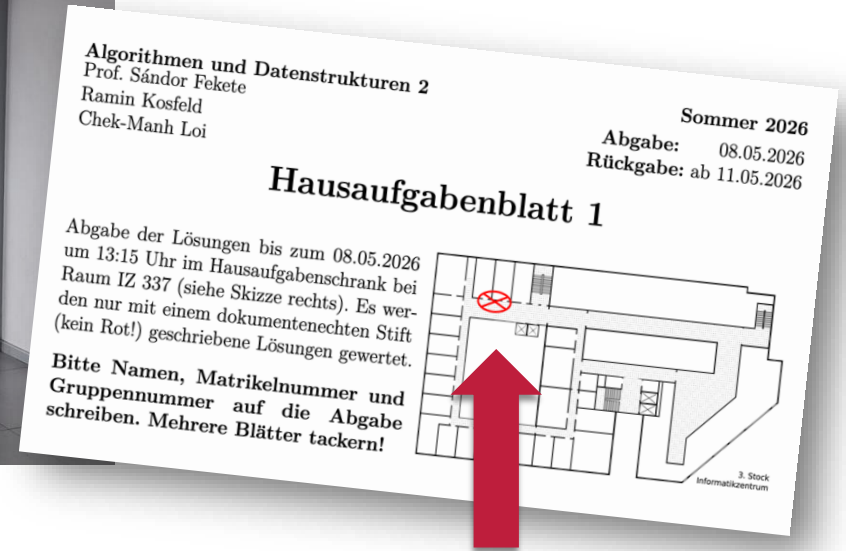
Abgabe der Lösungen bis zum 08.05.2026 um 13:15 Uhr im Hausaufgabenschrank bei Raum IZ 337 (siehe Skizze rechts). Es werden nur mit einem dokumentenechten Stift (kein Rot!) geschriebene Lösungen gewertet.

Bitte Namen, Matrikelnummer und Gruppennummer auf die Abgabe schreiben. Mehrere Blätter tackern!



Hausaufgaben

- Abgabe in Papierform
- Einwurf in HA-Schrank, 3. Stock IZ
- Vorsicht: Bis Freitag 13:15 Uhr!
 - Danach: Korrektur, aber keine Punkte.



Hausaufgabenschrank

Hausaufgaben

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 Lernstand

- Zusammen überlegen ist gut, aber:
*Einzel*n aufschreiben und abgeben.
Sonst: **Plagiat** = keine Punkte.

Homepages

TU Braunschweig

Was suchen Sie?

15

STUD.IP

Vorlesung/Übung: Algorithmen und Datenstrukturen - Details

Die Anmeldung zu dieser Veranstaltung folgt speziellen Regeln. Lesen Sie den Hinweistext.

Allgemeine Informationen

Veranstaltungsname	Vorlesung/Übung: Algorithmen und Datenstrukturen
Veranstaltungsnummer	4227001
Semester	WiSe 2023/24
Aktuelle Anzahl der Teilnehmenden	0
erwartete Teilnehmendenzahl	430
Heimat-Einrichtung	Abteilung Algorithmen (ALG)
Veranstaltungstyp	Vorlesung/Übung in der Kategorie Lehre

Aktionen

- Drucken
- Zugang zur Veranstaltung
- Nur im Stundenplan vormerken

Teilen

- Link zu dieser Veranstaltung kopieren

Stud.IP?



Homepages

<https://www.ibr.cs.tu-bs.de/courses/ss26/aud2/>

Algorithmen und Datenstrukturen 2

Semester Summer 2026 ▾

Module # INF-ALG-23


Event # INF-ALG-042, INF-ALG-043, INF-ALG-044

Programmes Business Information Systems Bachelor, Computer and Communication Systems Engineering Bachelor, Computer Science Bachelor


IBR Group ALG (Prof. Fekete)


Type Lecture & Exercise

Lecturer

 **Prof. Dr. Sándor P. Fekete**
Abteilungsleiter
✉ s.fekete@tu-bs.de
☎ +49 531 3913111
🏠 Room 335

Assistants

 **Ramin Kosfeld**
Wissenschaftlicher Mitarbeiter
✉ kosfeld@ibr.cs.tu-bs.de
☎ +49 531 3913119
🏠 Room 319

 **Chek-Manh Loi**
Wissenschaftlicher Mitarbeiter
✉ loi@ibr.cs.tu-bs.de
☎ +49 531 3913118
🏠 Room 331



Hier gibt es wirklich nichts zu sehen :)

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

The screenshot shows the course website for 'Algorithmen und Datenstrukturen 2' in the summer semester 2026. The header is red with the course title and semester. Below the header is a navigation menu with links for 'Startseite', 'Vorlesungen', 'Kleine Übung & Hausaufgaben', 'Kontakt', and 'Archiv'. The main content area is divided into two columns. The left column contains the course title, a description of the course as a mandatory event for students in computer science, and a list of topics: 'Elementare Aspekte zu Heuristiken', 'Exakte Verfahren: Dynamic Programming, Branch-and-Bound', and 'Approximationsalgorithmen'. The right column contains sections for 'Organisation', 'Vorlesung' (with dates and times), 'Große Übung' (with dates and times), and 'Klausur' (with dates and times).

Dort findet ihr

- Aktuelle Informationen
- Vorlesungsfolien
- Folien der Übungen (ja, auch die hier!)
- Alte Vorlesungsvideos (noch aus Corona, gut zum Wiederholen)
- Räume & Zeiten der Übungsgruppen
- Hausaufgaben
- Skript (bei Fehlern bitte Mail an uns)
- Altklausuren & alte Kursseiten
- Literaturempfehlungen
- Weiterführende Links

Semesterplan

Semesterplan (hier klicken)

Hier gibt es eine vollständige und stets aktuelle Übersicht über alle Termine der Veranstaltung.

Mailingliste (hier klicken)

Bitte tragt euch hier ein! Hier werden spontane Änderungen bekanntgegeben!

Semesterplan AuD II SS26

Woche (KW)	Woche (Datum)	Vorlesung (Di.)	Gr. Übung (Mi.)	Kl. Übung	HA Ausgabe	HA Abgabe bis Fr, 13:15	HA und Präsenz Besprechung
15	06.04.	/					
16	13.04.	1	0				
17	20.04.	2	1		HA1		
18	27.04.	3		0			P1
19	04.05.	4	2		HA2	HA1	
20	11.05.	5		1			HA1+P2
21	18.05.	6	3		HA3	HA2	
22	25.05.				Exkursionswoche		
23	01.06.	/		2			HA2+P3
24	08.06.	7			HA4	HA3	
25	15.06.	8	4	3			HA3+P4
26	22.06.	9			HA5	HA4	
27	29.06.	10	5	4			HA4+P5
28	06.07.	11				HA5	
29	13.07.	12	6	5			HA5+P6

Offizielle Übersicht über alle Termine, halten wir stets aktuell :)

Mailingliste

Aud2 aud2@ibr.cs.tu-bs.de

Summary

To contact the list owners, use the following email address: aud2-owner@ibr.cs.tu-bs.de
You have to login to visit the archives of this list.

Subscription / Unsubscription

To subscribe or unsubscribe from this list, please log in first. If you have not previously logged in, you may need to set up an account with the appropriate email address.

[Log in](#)

You can also subscribe without creating an account. If you wish to do so, please use the form below.

Your email address

Your name (optional)

[Subscribe](#)

- Anmeldung über Kursseite
- Kurzfristige Information zur Vorlesung/Übung/Klausur/etc.
- Bietet Möglichkeit, Fragen zu stellen (Aber wird an alle geschickt)

→ Wichtig, dass ihr da drauf seid!

Kleine Übung – Anmeldung

Anmeldung Kleine Übungen

/ Allgemein / Von Chek-Manh Loi

Die Anmeldung für c
Nach Anmeldeschlu
und Gruppe.

Weitere Infor

- Bitte wählt so vie
garantieren könn
- Wir werten immer
abschicken, wen
- Bei erfolgreicher
Daten. Das ist no
- Überprüft **vor** dem Absenden des Formulars, ob eure Daten (vor allem Matrikelnummer und eMail) korrekt sind.
- Bitte gebt als E-Mail-Adresse eure **TU-BS** Adresse an.
- Sollten unvorhergesehene Probleme auftreten, meldet euch bei [uns](#).

Nachname

Termine

Dienstag 13:15 Uhr

Ja

Mittwoch 16:45 Uhr

✓ Ja

Nein

Wenn Sein Muss

Nachname

Vorname

**Deadline: 19.04.2026
(Diesen Sonntag!)**

Bei Fragen ...

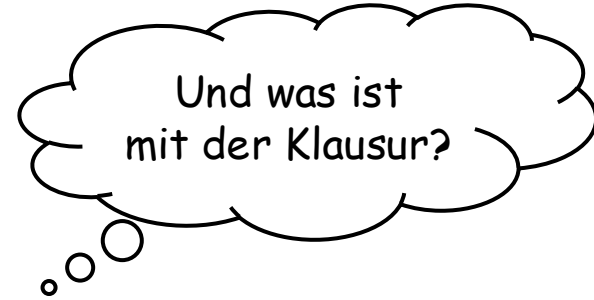
Bei Fragen zu den kleinen Übungen / Hausaufgaben etc.

- Fragt gerne zuerst bei euren Tutoren nach

Bei allgemeine Fragen, Organisatorisches etc.

- Sprecht uns nach der großen Übung an
- Schickt uns eine Mail
- Vereinbart einen Termin mit uns (per Mail)

Fragen?



Klausur

- Voraussichtlich 07.08.2026 um 10:30-12:30 Uhr.
- Details folgen irgendwann auf Webseite, gegen Semesterende
- Inhalt: Prinzipiell alles aus VL und Übung
- Dauer: 2 Stunden



Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmenik

Sommersemester 2024

Prof. Dr. Sándor Fekete
Ramin Kosfeld
Chek-Manh Loi

Klausur
Algorithmen und Datenstrukturen II
02.08.2024

Name:

Klausurcode:

Vorname:

Dieser wird benötigt, um das Ergebnis der Klausur abzurufen.

Matr.-Nr.:

Studiengang:

Bachelor Master Andere

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*

Laufzeit

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

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

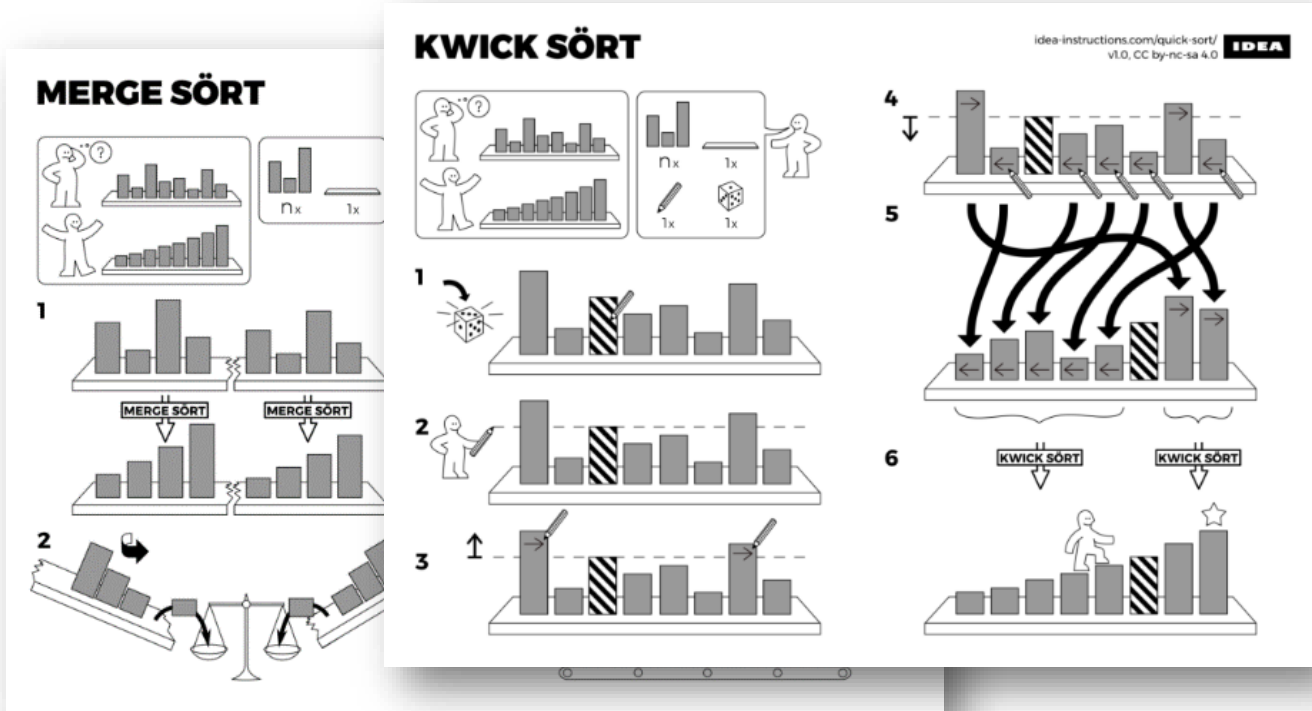
Können wir garantieren, dass

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

ab einem n_0 gilt, so schreiben wir

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

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	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$

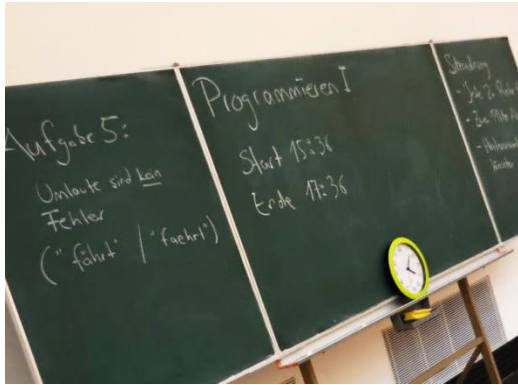
Theorem:

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

Und die schaffen wir ja auch, wie oben steht.

Gestern in der VL ...

Klausursituation



Verschiedene Klausuraufgaben ...

- Geben unterschiedlich viele Punkte
- Brauchen unterschiedlich viel Zeit
- Momentan: Teilpunkte möglich (= fraktionale Variante)

- Wie viele Punkte können wir maximal bekommen?

Gegeben:

n Objekte:

- p_i Punkte
- z_i Zeit

Zeitschranke Z

Gesucht:

Anteil $x_i \in [0,1]$, zu dem wir
jede Aufgabe bearbeiten

Lösungsansätze...

Berechne Verhältnis aus Zeit und Punkten

Sortiere nach Verhältnis

Wähle Aufgaben in absteigender Reihenfolge

Packe die letzte, passende Aufgabe anteilig,
wenn sie nicht mehr ganz passt



Gegeben:

n Objekte:

- p_i Punkte
- z_i Zeit

Zeitschranke Z

Gesucht:

Anteil $x_i \in [0,1]$, zu dem wir jede Aufgabe bearbeiten



Lösungsansätze...

Berechne Verhältnis
aus Zeit und Punkten

Sortiere nach Verhältnis

Wähle Aufgaben in
absteigender Reihenfolge

Packe die letzte, passende
Aufgabe anteilig, wenn sie
nicht mehr ganz passt

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**

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

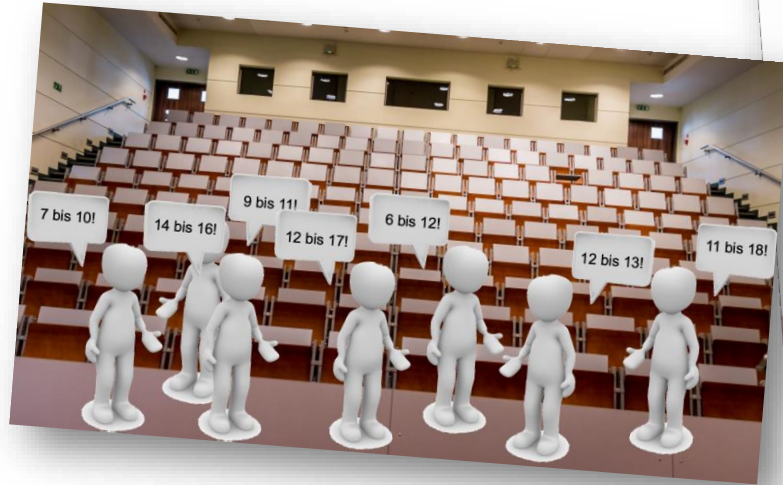
i	1	2	3	4	5
z_i	2	1	7	4	2
p_i	3	1	4	5	2
$\frac{z_i}{p_i}$	$\frac{2}{3}$	1	$\frac{7}{4}$	$\frac{4}{5}$	1
RF	1	3	5	2	4

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

Fragen?

... und nächstes Mal?

Nächste Woche :)



Beweis

Annahme: Algorithmus 1 sei nicht optimal.

Sei S_{alg} die Lösung von Algorithmus 1 und S_{opt} eine optimale Lösung.

Hier sind S_{alg} und S_{opt} identisch

Welche davon können das erste Intervall aus S_{opt} nicht in S_{alg} sein?

Dieser Bereich interessiert uns nicht

Erstes Intervall aus S_{alg} nicht in S_{opt} .

Technische Universität Braunschweig
Ramin Kosfeld und Chek-Manh Loi | Übung #1 | Hörsaal-Belegung | Seite 16

... wir basteln Raumpläne!