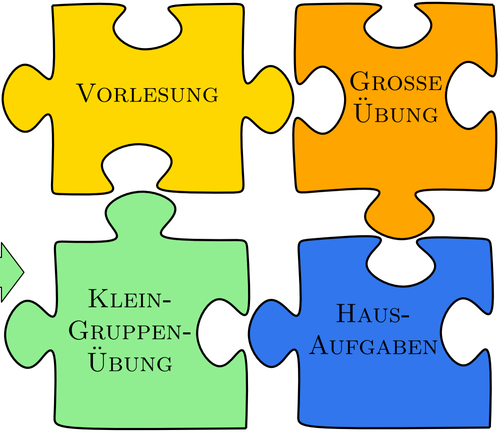
Technische
Universität
Braunschweig

Algorithmen und Datenstrukturen II

1. Vorlesung

Linda Kleist, 17.04.2019

Organisatorisches



Anmeldung

bis
24.04.2019,
23:59 Uhr

www.ibr.cs.tu-bs.de/courses/ss19/aud2/index.html

Organisatorisches

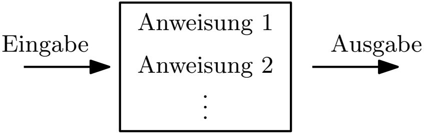
Semesterplan AuD II SS19

Woche	Vorlesung (Mi.)	Gr. Übung (Do.)	Kl. Übung (Mi./Fr.)	HA Ausgabe (Do.)	HA Abgabe (Do. bis 13:15 Uhr)	HA Rückgabe (in kl UE)
15	-					
16	1	0		HA0*		
17	2	1		HA1		
18	Tag der Arbeit		0**			HA0
19	3	2	(0**)	HA2	HA1	
20	4		1			HA1
21	5	3		HA3	HA2	
22	6	Himmelfahrt				
23	7	4	2	HA4	HA3	HA2
24	Exkursionswoche					
25	8		3			HA3
26	9	5		HA5	HA4	
27	10		4			HA4
28	11	6			HA5	
29	12	TDSE	5			HA5
*: Mündliche Aufgaben, Bearbeitung in ersten kleinen Übungen; keine Bewertung						
**: Da Mittwoch, der 01.05.19, ein Feiertag ist, finden die Mittwochsgruppen am 08.05.19 statt.						

Algorithmen & Datenstrukturen

Algorithmen

formale Handlungsvorschriften zur Lösung von Instanzen einer bestimmten Problemklasse.



- Vorschriften zur Verknüpfung zweier Zahlen $(+, -, \cdot, /)$
- Kochrezepte
- Sortieralgorithmen (AuD I)

Datenstrukturen

Konzepte zur Speicherung und Verwaltung von Daten, sowie zum Zugriff auf Daten.

- Stapel
- Warteschlangen
- Listen

Algorithmen & Datenstrukturen II – Inhalte

algorithmische Techniken

- Greedy
- dynamische Programmierung
- Branch & Bound
- Approximationsalgorithmen
- ...

weiterführende Datenstrukturen

- Hashing
- ...

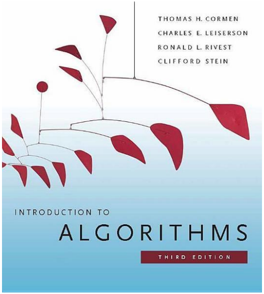
Komplexitäts-Theorie

leicht und *schwer*?

Technische Universität Braunschweig

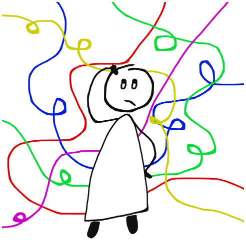
Algorithmen & Datenstrukturen II – Inhalte

Literaturhinweise



Einführendes Beispiel: Klausursituation

- Oskar sitzt in einer Klausur
- Bearbeitungszeit: 150 Minuten
- Maximalpunktzahl: 100 Punkte
- Bestehensgrenze: 50 Punkte



Situation nach 30 Minuten

- 6 Punkte sicher
- 10 Punkte hoffnungslos
- Überblick der restlichen Aufgaben

i Aufgabe	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
z_i Zeit	20	32	40	8	16	4	32	40	8	32	28	20	16	20	40	24
p_i Punkte	3	3	10	5	2	4	2	9	2	5	3	9	10	3	10	4

Kann Oskar die Klausur bestehen?

Einführendes Beispiel: Klausursituation

Kann Oskar die Klausur bestehen?

- verbleibende Zeit: 120 Minuten
- benötigte Punkte: ≥ 44

i Aufgabe	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
z_i Zeit	20	32	40	8	16	4	32	40	8	32	28	20	16	20	40	24
p_i Punkte	3	3	10	5	2	4	2	9	2	5	3	9	10	3	10	4

Gesucht:

Menge $S \subseteq \{1, 2, \ldots, 16\}$ mit

$$\sum_{i \in S} z_i \leq 120$$

$$\sum_{i \in S} p_i \geq 44$$

Einführendes Beispiel: Klausursituation

Tabelle, sortiert nach wertvollen Aufgaben:

i	6	4	13	12	9	3	15	8	16	10	1	14	5	11	2	7
z_i	4	8	16	20	8	40	40	40	24	32	20	20	16	28	32	32
p_i	4	5	10	9	2	10	10	9	4	5	3	3	2	3	3	2
$z_i/p_i \approx$	1.0	1.6	1.6	2.2	4.0	4.0	4.0	4.4	6	6.4	6.7	6.7	8.0	9.3	10.7	16.0

Vorgehen: Wähle die wertvollsten Aufgaben, die die Zeitschranke erlaubt:

$$S = \{6,$$

$$\sum_{i \in S} z_i = 4$$

$$\sum_{i \in S} p_i = 4$$

Einführendes Beispiel: Klausursituation

Tabelle, sortiert nach wertvollen Aufgaben:

i	6	4	13	12	9	3	15	8	16	10	1	14	5	11	2	7
z_i	4	8	16	20	8	40	40	40	24	32	20	20	16	28	32	32
p_i	4	5	10	9	2	10	10	9	4	5	3	3	2	3	3	2
$z_i/p_i \approx$	1.0	1.6	1.6	2.2	4.0	4.0	4.0	4.4	6	6.4	6.7	6.7	8.0	9.3	10.7	16.0

Vorgehen: Wähle die wertvollsten Aufgaben, die die Zeitschranke erlaubt:

$$S = \{6, 4$$

$$\sum_{i \in S} z_i = 12$$

$$\sum_{i \in S} p_i = 9$$

Einführendes Beispiel: Klausursituation

Tabelle, sortiert nach wertvollen Aufgaben:

i	6	4	13	12	9	3	15	8	16	10	1	14	5	11	2	7
z_i	4	8	16	20	8	40	40	40	24	32	20	20	16	28	32	32
p_i	4	5	10	9	2	10	10	9	4	5	3	3	2	3	3	2
$z_i/p_i \approx$	1.0	1.6	1.6	2.2	4.0	4.0	4.0	4.4	6	6.4	6.7	6.7	8.0	9.3	10.7	16.0

Vorgehen: Wähle die wertvollsten Aufgaben, die die Zeitschranke erlaubt:

$$S = \{6, 4, 13, 12, 9, 3, 15$$

$$\sum_{i \in S} z_i = 96 + 40 > 120$$

$$\sum_{i \in S} p_i = 40$$

Einführendes Beispiel: Klausursituation

Tabelle, sortiert nach wertvollen Aufgaben:

i	6	4	13	12	9	3	15	8	16	10	1	14	5	11	2	7
z_i	4	8	16	20	8	40	40	40	24	32	20	20	16	28	32	32
p_i	4	5	10	9	2	10	10	9	4	5	3	3	2	3	3	2
$z_i/p_i \approx$	1.0	1.6	1.6	2.2	4.0	4.0	4.0	4.4	6	6.4	6.7	6.7	8.0	9.3	10.7	16.0

Vorgehen: Wähle die wertvollsten Aufgaben, die die Zeitschranke erlaubt:

$$S = \{6, 4, 13, 12, 9, 3, 8$$

$$\sum_{i \in S} z_i = 96 + 40 > 120$$

$$\sum_{i \in S} p_i = 40$$

Einführendes Beispiel: Klausursituation

Tabelle, sortiert nach wertvollen Aufgaben:

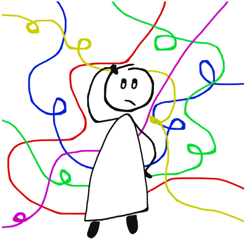
i	6	4	13	12	9	3	15	8	16	10	1	14	5	11	2	7
z_i	4	8	16	20	8	40	40	40	24	32	20	20	16	28	32	32
p_i	4	5	10	9	2	10	10	9	4	5	3	3	2	3	3	2
$z_i/p_i \approx$	1.0	1.6	1.6	2.2	4.0	4.0	4.0	4.4	6	6.4	6.7	6.7	8.0	9.3	10.7	16.0

Vorgehen: Wähle die wertvollsten Aufgaben, die die Zeitschranke erlaubt:

$$S = \{6, 4, 13, 12, 9, 3, 16\}$$

$$\sum_{i \in S} z_i = 120$$

$$\sum_{i \in S} p_i = 44$$



...Kann Oskar die Klausur bestehen?

Beschreibungen des Rucksackproblems

Beschreibungen des Rucksackproblems

Entscheidungsproblem

Problem 1 (0-1-Knapsack)

Gegeben:
- Objekte $\{O_1, \ldots, O_n\}$, je mit Größe z_i und Gewinn p_i
- Größenschranke Z
- Gewinnschranke P

Gesucht: Menge $S \subseteq \{1, \ldots, n\}$ mit

$$\sum_{i \in S} z_i \leq Z \text{ und}$$

$$\sum_{i \in S} p_i \geq P.$$

Beschreibungen des Rucksackproblems

Maximierungsproblem

Problem 2 (Maximum Knapsack)

Gegeben: - Objekte $\{O_1, \ldots, O_n\}$, je mit Größe z_i und Gewinn p_i
- Größenschranke Z

Gesucht: Menge $S \subseteq \{1, \ldots, n\}$ mit

$$\sum_{i \in S} z_i \leq Z \text{ und}$$

$$\sum_{i \in S} p_i \text{ maximal.}$$

Beschreibungen des Rucksackproblems

Problem 3 (Fractional Knapsack)

Teilpunkte

Gegeben:
- Objekte $\{O_1, \ldots, O_n\}$, je mit Größe z_i und Gewinn p_i
- Größenschranke Z

Gesucht: Ein Wert $x_i \in [0, 1]$ für jedes Objekt O_i, so dass

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

$$\sum_{i=1}^{n} x_i \cdot p_i \text{ maximal.}$$

Lösen von Fractional Knapsack

Algorithmus 1 (Greedy-Algorithmus für FRACTIONAL KNAPSACK)

Eingabe: $z_1, z_2, \ldots, z_n, Z, p_1, p_2, \ldots, p_n$

Ausgabe: $x_1, x_2, \ldots, x_n \in [0, 1]$ mit $\sum_{i=1}^{n} x_i \cdot z_i \leq Z$ und $\sum_{i=1}^{n} x_i \cdot p_i$ maximal.

1: sortiere $\{1, \ldots, n\}$ nach $\frac{z_i}{p_i}$ aufsteigend $\rightarrow$ Permutation $\pi(1), \ldots, \pi(n)$
2: **for** $k = 1$ to n **do**
3:     **if** $\left(\sum_{i=1}^{k} z_{\pi(i)} \leq Z\right)$ **then**
4:         $x_{\pi(k)} := 1$
5:     **else**
6:         $x_{\pi(k)} := \dfrac{Z - \sum_{i=1}^{k-1} x_{\pi(i)} z_{\pi(i)}}{z_{\pi(k)}}$
7: **return** $x_1, x_2, \ldots, x_n$