

Dynamic Programming

am Beispiel des Rucksackproblems

Problemstellung: Ein Rucksack habe ein Fassungsvermögen von K . Welche Items (Wert v_i , Gewicht w_i) sollte man einpacken, damit der Gesamtwert aller Gegenstände im Rucksack **maximal** wird?

Beispiel:



Mathematische Formulierung:

Maximiere $V = 1x_1 + 1x_2 + 1x_3 + 7x_4 + 10x_5 + 13x_6 + 10x_7$ (V = Gesamtwert)
 unter der Bedingung $G = 2x_1 + 2x_2 + 2x_3 + 3x_4 + 5x_5 + 8x_6 + 5x_7 \leq 10$ (W = Gesamtgewicht)
 mit $x_i \in \{0, 1\} \quad i \in \{1, 2, \dots, 7\}$

Die Lösung ist hier: $\vec{x} = (0, 0, 0, 0, 1, 0, 1)$ mit $V = 20$ und $W = 10$

Ein **Brute Force** Algorithmus müsste sämtliche Teilmengen überprüfen, d.h. 2^7 Lösungen.

Allgemein: n Items, mögliche Teilmengen: 2^n

Dynamic Programming:

Löse das Optimierungsproblem durch **Aufteilung in Teilprobleme** und systematische **Speicherung von Zwischenresultaten**.

Hier:

Löse das Problem für **null** Items und Kapazität **0**
 Löse das Problem für **null** Items und Kapazität **1**
 Löse das Problem für **null** Items und Kapazität **2**
 ...
 Löse das Problem für **null** Items und Kapazität **10**

 Löse das Problem für **ein** Item und Kapazität **0**
 Löse das Problem für **ein** Item und Kapazität **1**
 ...
 Löse das Problem für **ein** Item und Kapazität **10**
 ...

Löse das Problem für **zwei** Items und Kapazität **0**
 Löse das Problem für **zwei** Items und Kapazität **1**
 ...
 Löse das Problem für **zwei** Items und Kapazität **10**
 ...
 Löse das Problem für **sieben** Items und Kapazität **9**
 Löse das Problem für **sieben** Items und Kapazität **10**

Übungen

a)

<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">5 €</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">4 kg</div> <div>Item 1</div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">4 €</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">5 kg</div> <div>Item 2</div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">3 €</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">2 kg</div> <div>Item 3</div>	<div style="border: 1px solid black; border-radius: 15px; padding: 10px; background-color: #d1c4e9;"> Rucksack fasst 9 kg </div>
--	--	--	---

mathematisch:

maximize $5x_1 + 4x_2 + 3x_3$
subject to
 $4x_1 + 5x_2 + 2x_3 \leq 9$
 $x_i \in \{0, 1\} \quad (i \in 1..3)$

Capacity \ Item	0	1 <small>v₁ = w₁ =</small>	2 <small>v₂ = w₂ =</small>	3 <small>v₃ = w₃ =</small>
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				

Auszuwählen sind die Items _____ ,
der Gesamtwert der Items im Rucksack beträgt dann: _____

- b) maximize $16x_1 + 19x_2 + 23x_3 + 28x_4$
 subject to $2x_1 + 3x_2 + 4x_3 + 5x_4 \leq 7$
 $x_i \in \{0, 1\} \quad (i \in 1..4)$

Capacity \ Item	0	1 <small>v₁ = w₁ =</small>	2 <small>v₂ = w₂ =</small>	3 <small>v₃ = w₃ =</small>	4 <small>v₄ = w₄ =</small>
0					
1					
2					
3					
4					
5					
6					
7					

Auszuwählen sind die Items _____ ,
 der Gesamtwert der Items im Rucksack beträgt dann: _____

- c) Schreibe ein Computerprogramm, dass die im Kursweb hinterlegten Rucksack-Probleme löst.

Aufbau der Dateien (z.B. "ks_4_0"):

```

4 11          # Anzahl der Items in der Datei, Kapazität des Rucksacks
8 4           # Item 1: Wert, Gewicht
10 5          # Item 2...
15 8
4 3
```

Ausgabe deines Programms:

```

19            # Gesamtwert der augew. Items
0 0 1 1       # Lösungsvektor (hier: Item 3 und 4 werden ausgewählt)
```