Optimierung für Studierende der Informatik

Thomas Andreae

Wintersemester 2017/18
Blatt 12

A: Präsenzaufgaben am 22./23. Januar 2018

1. Erläutern Sie, wie die Einträge in der Tabelle im Beispiel auf Seite 206 zustande kommen.

Wie die unterste Zeile zustande kommt ist klar („Initialisierung").

Zu Zeile 1: Es gilt $w_1 = 2$. Die Einträge werden von links nach rechts vorgenommen, d.h., man betrachtet die Fälle $w=0, w=1, \ldots, w=6=W$. Für $w=0$ und $w=1$ hat man $w < w_1$, also gilt $M[1,0] = M[1,1] = 0$ (siehe erste Zeile von (13.3)). Für die restlichen Einträge ergibt sich (nach (13.3)) $M[1,w] = 2$ $(w = 2, \ldots, 6)$.

Zu Zeile 2: Es gilt $w_2 = 2$. Aufgrund von (13.3) erhält man $M[2,0] = M[2,1] = 0$, $M[2,2] = \max(M[1,2], 2+M[1,0]) = 2$, $M[2,3] = \max(M[1,3], 2+M[1,1]) = 2$, $M[2,4] = \max(M[1,4], 2+M[1,2]) = 4$; analog: $M[2,5] = M[2,6] = 4$.

Zu Zeile 3: $w_3 = 3 \Longrightarrow M[3,w] = M[2,w]$ für $w = 0,1,2$ sowie $M[3,3] = \max(M[2,3], 3+M[2,0]) = 3$,

$$M[3,4] = \max(M[2,4], 3+M[2,1]) = 4,$$
$$M[3,5] = \max(M[2,5], 3+M[2,2]) = 5,$$
$$M[3,6] = \max(M[2,6], 3+M[2,3]) = 5.$$

2. In Kapitel 6 des Skripts findet sich auf Seite 64 der folgende Text:

Knapsack

During a robbery, a burglar finds much more loot than he had expected and has to decide what to take. His bag (or „knapsack“) will hold a total weight of at most W pounds. There are n items to pick from, of weight $w_1, \ldots, w_n$ and dollar value $v_1, \ldots, v_n$. What's the most valuable combination of items he can fit into his bag?

For instance, take $W = 10$ and

Item	1	2	3	4
Weight	6	3	4	2
Value	\$30	\$14	\$16	\$9

a) Wir betrachten die Variante, in der jeder Gegenstand nur einmal vorhanden ist. Lösen Sie das Problem für die angegebenen Daten, indem Sie den auf Seite 207 beschriebenen Dynamischen-Programmierungs-Algorithmus verwenden.
Hinweis: Es ist eine Tabelle anzulegen, die der Tabelle aus Aufgabe 1 sehr ähnlich ist.

b) Wie kann man aus der Tabelle nicht nur den optimalen Wert einer Rucksackfüllung ablesen, sondern auch, *welche Gegenstände* in den Rucksack zu packen sind?

---

a) Es geht alles analog zu Aufgabe 1, man muss nur darauf achten, dass nicht mehr $v_i = w_i$ gilt. Man erhält:

$w_4=2, v_4=9$	4	0	0	9	14	16	23	30	30	39	44	46
$w_3=4, v_3=16$	3	0	0	0	14	16	16	30	30	30	44	46
$w_2=3, v_2=14$	2	0	0	0	14	14	14	30	30	30	44	44
$w_1=6, v_1=30$	1	0	0	0	0	0	0	30	30	30	30	30
	0	0	0	0	0	0	0	0	0	0	0	0
		0	1	2	3	4	5	6	7	8	9	10

Legt man die Tabelle mit Hilfe der Rekursionsformel auf Seite 207 des Skripts an, so ist es zweckmäßig, die Gewichte w_i und Werte v_i direkt neben der entsprechenden Zeile stehen zu haben – zumindest, wenn man an der Tafel arbeitet.

b) <u>Man beginnt rechts oben mit dem optimalen Wert einer Rucksackfüllung (im Beispiel: 46) und verfolgt zurück, wie dieser Wert zustande gekommen ist.</u>
In unserem Beispiel läuft das so ab:

Da in der letzten Spalte unterhalb der rechts oben stehenden 46 wiederum 46 steht, kommt Gegenstand 4 nicht in den Rucksack. Unmittelbar darunter taucht in der letzten Spalte ein Eintrag auf, der verschieden von 46 ist (nämlich 44): Also wandert Gegenstand 3 in den Rucksack. Da Gegenstand 3 das Gewicht 4 hat, geht man in der Zeile, die zum Gegenstand 2 gehört, vier Spalten nach links, wo man auf den Eintrag 30 stößt. Unterhalb dieser 30 steht ebenfalls der Eintrag 30, weshalb Gegenstand 2 nicht in den Rucksack kommt. Unter der letztgenannten 30 findet man einen kleineren Eintrag als 30 (nämlich 0), weshalb Gegenstand 1 in den Rucksack gepackt wird.

<u>Ergebnis</u>: Gegenstand 1 und 3 kommen in den Rucksack.

In der folgenden Darstellung wurden die Einträge unterstrichen, auf die es bei der Bestimmung der optimalen Rucksackfüllung ankam (siehe auch Hausaufgabe 1):

4	0	0	9	14	16	23	30	30	39	44	46
3	0	0	0	14	16	16	30	30	30	44	46
2	0	0	0	14	14	14	30	30	30	44	44
1	0	0	0	0	0	0	30	30	30	30	30
0	0	0	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9	10

3. *Eine Klausuraufgabe aus dem WS 2013/14*: Wir betrachten die Variante des Rucksackproblems, bei der jeder Gegenstand nur einmal vorhanden ist. Die Gegenstände bezeichnen wir mit $1, \ldots, n$. Mit v_i sei der Wert („value") des Gegenstandes i bezeichnet und w_i bezeichne sein Gewicht („weight"). Für jeden Gegenstand betrachten wir den Quotienten $q_i = \frac{v_i}{w_i}$ („Wert einer Gewichtseinheit") und nehmen an, dass die Quotienten alle verschieden sind.

Vorschlag für eine Greedy-Strategie: Man ordnet die Gegenstände in absteigender Reihenfolge nach den Quotienten („Gegenstand mit größtem Quotienten zuerst"). In dieser Reihenfolge geht man die Gegenstände durch und packt immer den nächsten noch möglichen Gegenstand ein. Beweisen oder widerlegen Sie die Behauptung, dass diese Strategie immer eine optimale Lösung liefert.

Schon bei $n = 3$ Gegenständen versagt diese Strategie, wie folgendes Beispiel zeigt:

Item	1	2	3
Weight	3	2	2
Value	8	5	4

Maximallast $W = 4$

B: Hausaufgaben zum 29./30. Januar 2018

1. Wie Präsenzaufgabe 2, aber diesmal für folgende Daten sowie $W = 18$:

Item	1	2	3	4	5	6	7
Weight	4	6	11	8	7	5	3
Value	2	3	6	6	5	4	2

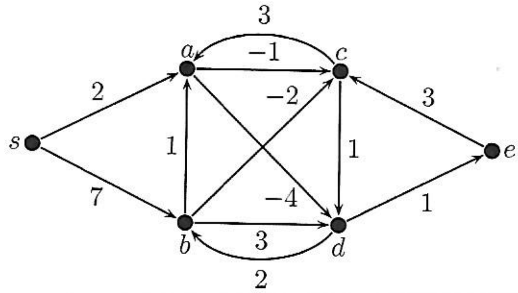
Es ist auch eine optimale Rucksackfüllung an der von Ihnen angelegten Tabelle abzulesen. Unterstreichen Sie diejenigen Einträge der Tabelle, auf die es beim Ablesen der optimalen Rucksackfüllung ankam, und geben Sie die gefundene Rucksackfüllung an.

7	0	0	0	2	2	4	4	5	6	6	7	8	9	10	10	11	12	12	13
6	0	0	0	0	2	4	4	5	6	6	6	7	9	10	10	11	11	12	12
5	0	0	0	0	2	2	3	5	6	6	6	7	8	8	9	11	11	11	11
4	0	0	0	0	2	2	3	3	6	6	6	6	8	8	9	9	9	9	11
3	0	0	0	0	2	2	3	3	3	3	5	6	6	6	6	8	8	9	9
2	0	0	0	0	2	2	3	3	3	3	5	5	5	5	5	5	5	5	5
1	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Man liest ab (siehe Unterstreichungen), dass die Gegenstände 7, 5 und 4 eine optimale Rucksackfüllung bilden

2. Wenden Sie auf die folgenden Graphen G die Version des Algorithmus von Bellman und Ford an, bei der man am Schluss feststellt, ob der gegebene Graph einen negativen Kreis enthält. Es ist eine *Tabelle* anzulegen, an der man erstens ablesen kann, ob ein negativer Kreis vorhanden ist; falls dies nicht der Fall ist, so soll man zweitens an der Tabelle für alle Knoten v sowohl die Länge eines kürzesten s, v-Pfades als auch einen solchen Pfad selber ablesen können.

a) G sei der folgende Graph:

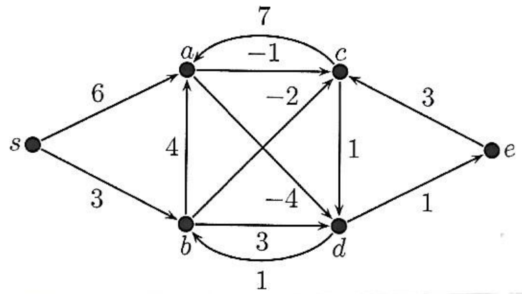


Für G erhält man folgende Tabelle:

	s	a	b	c	d	e
0	0 −	∞ −	∞ −	∞ −	∞ −	∞ −
1	0 −	2 s	7 s	∞ −	∞ −	∞ −
2	0 −	2 s	7 s	1 a	−2 a	∞ −
3	0 −	2 s	0 d	1 a	−2 a	−1 d
4	0 −	1 b	0 d	−2 b	−2 a	−1 d
5	0 −	1 b	0 d	−2 b	−3 a	−1 d
6	0 −	1 b	−1 d	−2 b	−3 a	−2 d

Da die vorletzte und die letzte Zeile nicht übereinstimmen, enthält G einen negativen Kreis.

b) Der Graph G sei wie folgt gegeben:



Man erhält die folgende Tabelle:

	s	a	b	c	d	e
0	0 –	∞ –	∞ –	∞ –	∞ –	∞ –
1	0 –	6 s	3 s	∞ –	∞ –	∞ –
2	0 –	6 s	3 s	1 b	2 a	∞ –
3	0 –	6 s	3 s	1 b	2 a	3 d
4	0 –	6 s	3 s	1 b	2 a	3 d
5	0 –	6 s	3 s	1 b	2 a	3 d
6	0 –	6 s	3 s	1 b	2 a	3 d

Hier noch zusätzlich ein Kürzester-Pfade-Baum, den man an der untersten Zeile der Tabelle ablesen kann: