



## Lösung 13: Dynamische Programmierung

### Aufgabe 1: Rod Cutting

- a) Der maximale Erlös  $r[9]$  beträgt 25, für  $s[9]$  ergibt sich 3. Am Ende der inneren Schleife (Zeile 8) ergeben sich jeweils die folgenden Werte:

Index i innere Schleife	q	s[9]
1	23	1
2	23	1
3	25	3
4	25	3
5	25	3
6	25	3
7	25	3
8	25	3
9	25	3

- b) Der maximale Erlös  $r[10]$  beträgt 30, für  $s[10]$  ergibt sich 10, man muss also gar nicht unterteilen. Am Ende der inneren Schleife (Zeile 8) ergeben sich jeweils die folgenden Werte:

Index i innere Schleife	q	s[9]
1	26	1
2	27	2
3	27	2
4	27	2
5	27	2
6	27	2
7	27	2
8	27	2
9	30	10

- c) Die asymptotische Laufzeit beträgt  $\Theta(n^2)$ . Falls der Stab an beliebigen Stellen unterteilbar wäre, könnte man keine dynamische Programmierung mehr verwenden, da die Tabelle unendlich groß werden würde. Die Berechnung würde also unendlich lange dauern.

## Aufgabe 2: Längste Gemeinsame Teilfolge / Longest Common Subsequence (LCS)

Bestimmen Sie die Längste Gemeinsame Teilfolge der folgenden beiden Folgen:

- $X = \langle B, A, A, B \rangle$  und
- $Y = \langle A, B, A, B, B, A, B \rangle$

	$j$	0	1	2	3	4	5	6	7
$i$		$y_i$	A	B	A	B	B	A	B
0	$x_i$	0	0	0	0	0	0	0	0
1	B	0	0 ↑	1 ↖	1 ←	1 ↖	1 ↖	1 ←	1 ↖
2	A	0	1 ↖	1 ↑	2 ↖	2 ←	2 ←	2 ↖	2 ←
3	A	0	1 ↖	1 ↑	2 ↖	2 ↑	2 ↑	3 ↖	3 ←
4	B	0	1 ↑	2 ↖	2 ↑	3 ↖	3 ↖	3 ↑	4 ↖

Die vom Algorithmus bestimmte Längste Gemeinsame Teilfolge ist also „B A A B“. Die komplette Sequenz X ist also in Y enthalten.

## Aufgabe 3: Levenshtein - Editierdistanz

- a) Ergänzen Sie die untenstehende Tabelle, um die Levenshtein-Distanz für die Überführung der Zeichenkette X in die Zeichenkette Y zu berechnen. Was ist die Editierdistanz?

	$j$	0	1	2	3	4	5	6	7	8	9	10	11
$i$		$y_i$	M	E	I	L	E	N	S	T	E	I	N
0	$x_i$	0	1	2	3	4	5	6	7	8	9	10	11
1	L	1	1	2	3	3	4	5	6	7	8	9	10
2	E	2	2	1	2	3	3	4	5	6	7	8	9
3	V	3	3	2	2	3	4	4	5	6	7	8	9
4	E	4	4	3	3	3	3	4	5	6	6	7	8
5	N	5	5	4	4	4	4	3	4	5	6	7	7
6	S	6	6	5	5	5	5	4	3	4	5	6	7
7	H	7	7	6	6	6	6	5	4	4	5	6	7
8	T	8	8	7	7	7	7	6	5	4	5	6	7
9	E	9	9	8	8	8	7	7	6	5	4	5	6
10	I	10	10	9	8	9	8	8	7	6	5	4	5
11	N	11	11	10	9	9	9	8	8	7	6	5	4

- b) Mögliche Implementierung, siehe Quelltext (kein Klausurstoff)

[https://inf-git.fh-](https://inf-git.fh-rosenheim.de/muwo522/ad_wise_2019/tree/master/src/de/th_rosenheim/ad/uebung13)

[rosenheim.de/muwo522/ad\\_wise\\_2019/tree/master/src/de/th\\_rosenheim/ad/uebung13](https://inf-git.fh-rosenheim.de/muwo522/ad_wise_2019/tree/master/src/de/th_rosenheim/ad/uebung13)

- c)  $O(mn)$

- d) Es gibt 2 mögliche Wege um durch die Tabelle zu kommen, siehe gelb markierte Felder aus Aufgabe a). Man beginnt rechts unten und geht immer zum Minimum aus „links, links oben und oben“ zurück. Diagonal nach links oben gehen bedeutet dabei „Übereinstimmung“ oder „Substitution“. Nach links gehen bedeutet dabei „Einfügen“, nach oben gehen bedeutet, dass eine „Löschen-Operation“ stattgefunden hat.

Im konkreten Beispiel gibt es 2 mögliche Wege mit den geringsten Kosten:

l	e		v	e	n	s	h	t	e	i	n
o	=	+	o	=	=	=	-	=	=	=	=
m	e	i	l	e	n	s		t	e	i	n

or

l	e	v		e	n	s	h	t	e	i	n
o	=	o	+	=	=	=	-	=	=	=	=
m	e	i	l	e	n	s		t	e	i	n

"=" Übereinstimmung; "o" Ersetzen; "+" Einfügen; "-" Löschen