TECHNISCHE
UNIVERSITÄT
DRESDEN

ALGORITHMEN UND DATENSTRUKTUREN

ÜBUNG 9: SUCHEN & ERSETZEN

Eric Kunze
`eric.kunze@tu-dresden.de`

TU Dresden, 1. Dezember 2021

letzte Änderung:
30.11.2021, 15:16

KMP-Algorithmus

Aufgabe 1

KMP-ALGORITHMUS

- Mustersuche in (großen) Texten
- Ziel: Verschiebung des Musters um mehr als eine Position bei Nichtübereinstimmung.
- Methode: Ermittlung einer Verschiebetabelle `Tab[]` in **Phase 1**
- Bedeutung des Eintrags `Tab[i]=j`:
  Bei Nichtübereinstimmung an Stelle i wird Position j des Musters an aktueller Vergleichsstelle angelegt.
- Suchprozess in **Phase 2**

j-algo: `http://j-algo.binaervarianz.de/`

KMP-ALGORITHMUS

Suche das Muster `aaabaaaa` im Text `aaabaaabaaacaaabaaaa`.

Position	0	1	2	3	4	5	6	7
Pattern	a	a	a	b	a	a	a	a
Tabelle	-1	-1	-1	2	-1	-1	-1	3

Erster Versuch:

a a a b a a a **b** a a a c a a a b a a a a
a a a b a a a **a**

Tabelleneintrag an Position 7 ist 3, d.h. `Tab[7]=3` — Lege Position 3 des Musters an aktueller Vergleichsposition an:

a a a b a a a b a a a **c** a a a b a a a a
a a a b a a a **a**

Gleicher Prozess noch einmal: Missmatch an Position 7 des Musters — verschiebe Muster auf Position 3.

KMP-ALGORITHMUS (FORTSETZUNG)

Suche das Muster `aaabaaaa` im Text `aaabaaabaaacaaabaaaa`.

Position	0	1	2	3	4	5	6	7
Pattern	a	a	a	b	a	a	a	a
Tabelle	-1	-1	-1	2	-1	-1	-1	3

Wir legen das Muster also wieder an Position 3 an:

a a a b a a a b a a a **c** a a a b a a a a
a a a **b** a a a a

Wegen `Tab[3]=2`, lege Muster an Position 2 an:

a a a b a a a b a a a **c** a a a b a a a a
a a **a** b a a a a

Wegen `Tab[2]=-1`, lege Muster an Position -1 an:

a a a b a a a b a a a c a a a b a a a a ☺
a a a b a a a a

KMP-ALGORITHMUS — DIE ZYKLENMETHODE

Zwei Phasen:

- **1. Phase:** Markieren der längsten Teilwörter im Pattern, die mit einem Präfix übereinstimmen
  - ein Zyklus beginnt an einer Patternposition `i` falls $i \neq 0$ und `Pat[0] = Pat[i]`
  - ein Zyklus endet an der kleisten Patternposition `i+m`, sodass `Pat[m+1]` $\neq$ `Pat[i+m+1]`
- **2. Phase:** Bestimmung der Tabelleneinträge
  - `Tab[0] = -1`
  - Tabelleneinträge nach einem Zyklus: *Länge des längsten dort endenden Zyklus*
  - Tabelleneinträgen in einem Zyklus: *Tabelleneintrag der derzeitigen Position im längsten laufenden Zyklus*
  - verbleibende Einträge: `0`

KMP-ALGORITHMUS — DIE ZWEI-FINGER-METHODE

Die Methode beruht auf der Gleichung

$$\texttt{Tab[i]} = \max\,\{-1\} \cup \left\{ m \;\middle|\; \begin{array}{c} 0 \leq \; m \; \leq i-1 \\ b_0 \ldots b_{m-i} \; = \; b_{i-m} \ldots b_{i-1} \\ b_m \; \neq \; b_j \end{array} \right\} \qquad (\star)$$

Daraus ergibt sich nach Initialisierung von `Tab[0]` $= -1$ für jeden folgenden Eintrag `Tab[i]` folgendes Verfahren:

- *linker Finger*: wähle $m < i$ in absteigender Reihenfolge (also $i-1$, $i-2$, $\ldots$), sodass `Pat[i]` $\neq$ `Pat[m]`
- *Parallelverschiebung beider Finger bis zum linken Rand*: wenn `Pat[0...m-1]` $=$ `Pat[i-m...i-1]`, dann fülle `Tab[i]` $= m$.
- wenn keine passende Position m gefunden werden kann, dann fülle `Tab[i]` $= -1$.

AUFGABE 1

LÖSUNG

Teil (a) Pattern: `aabaaacaab`

Position	0	1	2	3	4	5	6	7	8	9
Pattern	a	a	b	a	a	a	c	a	a	b
Tabelle	-1	-1	1	-1	-1	2	2	-1	-1	1

AUFGABE 1 — TEIL (B)

LÖSUNG

Teil (b)

Position	0	1	2	3	4	5
Pattern	c	b	*c*	*c*	*b*	a
Tabelle	-1	0	-1	1	0	2

- `Pat[0...1] = Pat[3...4]` wegen `Tab[5] = 2` (Zyklenmethode), d.h. `Pat[3] = Pat[0] = c` und `Pat[4] = Pat[1] = b`
- wegen `Tab[3] = 1` ist `Pat[2] = Pat[0] = c` (Zyklenmethode)
- **oder:** wegen `Tab[3] = 1` ist `Pat[1]` ≠ `Pat[3]` und `Pat[2] = Pat[0] = c` (Parallelverschiebung in der Zwei-Finger-Methode bzw. Gleichung (⋆))

Levenshtein-Distanz

Aufgabe 2

LEVENSHTEIN-DISTANZ

Kosten zur Überführung eines Wortes $w = w_1 \dots w_n$ in ein Wort $v = v_1 \dots v_k$; schreibe $d(w_1 \dots w_j, v_1 \dots v_i) = d(j, i)$.

$$d(0, i) = i$$

$$d(j, 0) = j$$

$$d(j, i) = \min \{d(j, i-1) + 1, d(j-1, i) + 1, d(j-1, i-1) + \delta_{j,i}\}$$

für alle $1 \leq j \leq n$ und alle $1 \leq i \leq k$ wobei

$$\delta_{j,i} = \begin{cases} 1 & \text{wenn } w_j \neq v_i \\ 0 & \text{sonst} \end{cases}$$

Anschaulich: Überlagerung durch Pattern → Pfeile zeigen ”Ursprung“ des Minimums an

$w_j \neq v_i$:

$+1$	$+1$
$+1$	?

$w_j = v_i$:

$+0$	$+1$
$+1$	?

AUFGABE 2

Gegeben seien die Wörter $w = \texttt{espen}$ und $v = \texttt{beispiele}$.

(a) Berechnen Sie die Levenshtein-Distanz $d(w, v)$. Geben Sie dazu die Berechnungsmatrix an. Tragen Sie alle Zelleneinträge zusammen mit den dazugehörigen Pfeilen ein.

(b) Geben Sie die Levenshtein-Distanz $d(\texttt{espe}, \texttt{beispiel})$ an. Beachten Sie, dass `espe` und `beispiel` Präfixe von `espen` bzw. `beispiele` sind.

(c) Geben Sie zwei Alignments zwischen `espen` und `beispiele` an, die zu den minimalen Kosten führen. Dabei sollen die Alignments die jeweils angewendeten Editieroperation enthalten.

(d) Wieviele Alignments enthält die in Aufgabe (a) angegebene Berechnungsmatrix?

AUFGABE 2

LÖSUNG

Teil (a) $d(\text{espen}, \text{beispiele}) = 5$

$$
\begin{array}{c|ccccccccccccccccccc}
d(j,i) & & & \mathbf{b} & & \mathbf{e} & & \mathbf{i} & & \mathbf{s} & & \mathbf{p} & & \mathbf{i} & & \mathbf{e} & & \mathbf{l} & & \mathbf{e} \\
\hline
 & 0 & \rightarrow & 1 & \rightarrow & 2 & \rightarrow & 3 & \rightarrow & 4 & \rightarrow & 5 & \rightarrow & 6 & \rightarrow & 7 & \rightarrow & 8 & \rightarrow & 9 \\
 & & \searrow & & \searrow & & & & & & & & & & \searrow & & & & \searrow & \\
\mathbf{e} & 1 & & 1 & & 1 & \rightarrow & 2 & \rightarrow & 3 & \rightarrow & 4 & \rightarrow & 5 & \rightarrow & 6 & \rightarrow & 7 & \rightarrow & 8 \\
 & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & & \searrow & & & & & & & & & & & \\
\mathbf{s} & 2 & & 2 & & 2 & & 2 & & 2 & \rightarrow & 3 & \rightarrow & 4 & \rightarrow & 5 & \rightarrow & 6 & \rightarrow & 7 \\
 & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & & & & & & & & & \\
\mathbf{p} & 3 & & 3 & & 3 & & 3 & & 3 & & 2 & \rightarrow & 3 & \rightarrow & 4 & \rightarrow & 5 & \rightarrow & 6 \\
 & \downarrow & \searrow & \downarrow & \searrow & & \searrow & \downarrow & \searrow & \downarrow & & \downarrow & \searrow & & \searrow & & & & \searrow & \\
\mathbf{e} & 4 & & 4 & & 3 & \rightarrow & 4 & & 4 & & 3 & & 3 & & 3 & \rightarrow & 4 & \rightarrow & 5 \\
 & \downarrow & \searrow & \downarrow & & \downarrow & \searrow & & \searrow & \downarrow & & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & & \searrow & \\
\mathbf{n} & 5 & & 5 & & 4 & & 4 & \rightarrow & 5 & & 4 & & 4 & & 4 & & 4 & \rightarrow & 5
\end{array}
$$

Teil (b) $d(\text{espe}, \text{beispiel}) = 4$

AUFGABE 2 LÖSUNG

Teil (c) Alignments mit minimaler Levenshtein-Distanz:

*	e	*	s	p	*	e	*	n
|	|	|	|	|	|	|	|	
b	e	i	s	p	i	e	l	e
i		*i*			*i*		*i*	*s*

*	e	*	s	p	*	e	n	*
|	|	|	|	|	|	|	|	
b	e	i	s	p	i	e	l	e
i		*i*			*i*		*s*	*i*

Teil (d) 2 Alignments = 2 Backtraces

Weitere Aufgaben aus der Aufgabensammlung *mit Lösungen*

AUFGABE 7.1.13 (AGS)

(a) Bestimmen Sie die mit Hilfe des KMP-Algorithmus berechnete Verschiebetabelle für das Pattern `abbababbaa`.

(b) Mit Hilfe des KMP-Algorithmus ist unten stehende Verschiebetabelle berechnet worden. Die mit einem „?“ markierten Einträge sind unbekannt. Vervollständigen Sie das aus den Symbolen `a`, `b` und `c` bestehende Pattern.

Position	0	1	2	3	4	5
Pattern	*b*					*c*
Tabelle	-1	?	?	0	?	3

AUFGABE 7.1.13 (AGS)

LÖSUNG

Teil (a) Pattern: `abbabbaa`

Position	0	1	2	3	4	5	6	7
Pattern	a	b	b	a	b	b	a	a
Tabelle	-1	0	0	-1	0	0	-1	4

Teil (b)

Position	0	1	2	3	4	5
Pattern	*b*	a	b	a	b	*c*
Tabelle	-1	?	?	0	?	3

- `Pat[0 ... 2] = Pat[2 ... 4]` wegen `Tab[5] = 3` (Zyklenmethode), d.h. `Pat[2] = Pat[0] = Pat[4] = b`
- wegen `Tab[3] = 0` ist `Pat[3]` $\neq$ `Pat[0] = b` und wegen `Tab[5] = 3` ist `Pat[3]` $\neq$ `Pat[5] = c` (Zwei-Finger-Methode bzw. Gleichung ($\star$)) $\Rightarrow$ `Pat[3] = Pat[1] = a`

AUFGABE 7.2.1 (AGS)

Gegeben seien die Wörter $w = \texttt{Dinstas}$ und $v = \texttt{Distanz}$.

(a) Berechnen Sie die Levenshtein-Distanz $d(w, v)$ zwischen w und v. Geben Sie die Berechnungsmatrix vollständig an.

(b) Geben Sie alle Alignments mit minimaler Levenshtein-Distanz zwischen w und v an.

AUFGABE 7.2.1 (AGS)

LÖSUNG

$d(j,i)$			D		i		s		t		a		n		z
	0	→	1	→	2	→	3	→	4	→	5	→	6	→	7
	↓	↘													
D	1		0	→	1	→	2	→	3	→	4	→	5	→	6
	↓		↓	↘											
i	2		1		0	→	1	→	2	→	3	→	4	→	5
	↓		↓		↓	↘		↘		↘		↘			
n	3		2		1		1	→	2	→	3		3	→	4
	↓		↓		↓	↘		↘		↘		↘	↓	↘	
s	4		3		2		1	→	2	→	3	→	4		4
	↓		↓		↓		↓	↘							
t	5		4		3		2		1	→	2	→	3	→	4
	↓		↓		↓		↓		↓	↘					
a	6		5		4		3		2		1	→	2	→	3
	↓		↓		↓	↘	↓		↓		↓	↘		↘	
s	7		6		5		4		3		2		2	→	3

$$d(\text{Dinstas}, \text{Distanz}) = 3$$

AUFGABE 7.2.1 (AGS)

LÖSUNG

Alignments mit minimaler Levenshtein-Distanz:

$$\begin{array}{cccccccc}
\mathrm{D} & \mathrm{i} & \mathrm{n} & \mathrm{s} & \mathrm{t} & \mathrm{a} & * & \mathrm{s} \\
| & | & | & | & | & | & | & | \\
\mathrm{D} & \mathrm{i} & * & \mathrm{s} & \mathrm{t} & \mathrm{a} & \mathrm{n} & \mathrm{z} \\
 & & d & & & & i & s
\end{array}$$

$$\begin{array}{cccccccc}
\mathrm{D} & \mathrm{i} & \mathrm{n} & \mathrm{s} & \mathrm{t} & \mathrm{a} & \mathrm{s} & * \\
| & | & | & | & | & | & | & | \\
\mathrm{D} & \mathrm{i} & * & \mathrm{s} & \mathrm{t} & \mathrm{a} & \mathrm{n} & \mathrm{z} \\
 & & d & & & & s & i
\end{array}$$

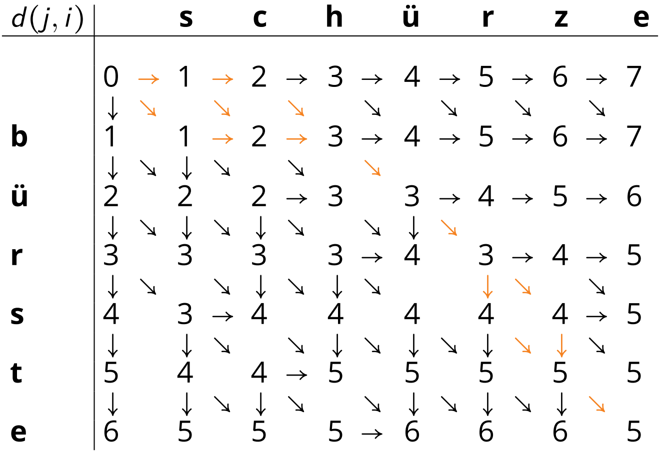
AUFGABE 7.2.2 (AGS)

(a) Berechnen Sie die Levenshtein-Distanz $d(\texttt{bürste}, \texttt{schürze})$. Geben Sie die Berechnungsmatrix vollständig an. Wieviele Backtraces enthält die Berechnungsmatrix?

(b) Geben Sie zwei Alignments mit minimaler Levenshtein-Distanz zwischen den Wörtern `bürst` und `sch` an.

AUFGABE 7.2.2 (AGS) — TEIL (A)

LÖSUNG



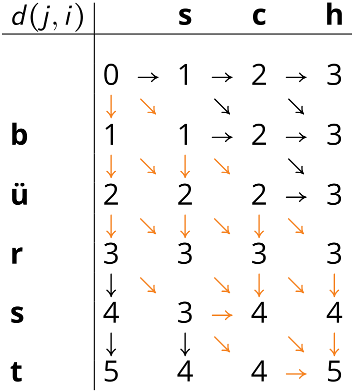
$d(j,i)$		s	c	h	ü	r	z	e
	0	1	2	3	4	5	6	7
b	1	1	2	3	4	5	6	7
ü	2	2	2	3	3	4	5	6
r	3	3	3	3	4	3	4	5
s	4	3	4	4	4	4	4	5
t	5	4	4	5	5	5	5	5
e	6	5	5	5	6	6	6	5

$d(\text{bürste}, \text{schürze}) = 5$ Anzahl der Backtraces $= 3 * 2 = 6$

AUFGABE 7.2.2 (AGS) — TEIL (B)

LÖSUNG

Alignments mit minimaler Levenshtein-Distanz zwischen den Wörtern `bürst` und `sch`



$d(j, i)$			**s**		**c**		**h**
	0	→	1	→	2	→	3
	↓	↘		↘		↘	
b	1		1	→	2	→	3
	↓	↘	↓	↘		↘	
ü	2		2		2	→	3
	↓	↘	↓	↘	↓	↘	
r	3		3		3		3
	↓	↘		↘	↓	↘	↓
s	4		3	→	4		4
	↓		↓	↘		↘	↓
t	5		4		4	→	5

AUFGABE 7.2.2 (AGS) — TEIL (B)

LÖSUNG

Alignments mit minimaler Levenshtein-Distanz zwischen den Wörtern `bürst` und `sch`

b	ü	r	s	t
|	|	|	|	|
s	c	h	*	*
s	*s*	*s*	*d*	*d*

b	ü	r	s	t
|	|	|	|	|
*	*	s	c	h
d	*d*	*s*	*s*	*s*