TECHNISCHE UNIVERSITÄT DRESDEN

ALGORITHMEN UND DATENSTRUKTUREN

ÜBUNG 10: SUCHEN & KORRIGIEREN, AVL-BÄUME

Eric Kunze

eric.kunze@mailbox.tu-dresden.de

TU Dresden, 09.01.2020

Was bisher geschah ...

- Syntax von Programmiersprachen (Syntaxdiagramme, EBNF, Fixpunktsemantik)
- Programmieren in *C* – Arrays, Pointer, Listen, Bäume
- grundlegende Algorithmen in der Informatik
  - Sortieren mit Quicksort und Heapsort
  - Suchen in Texten (KMP-Algorithmus)

Was bisher geschah ...

- Syntax von Programmiersprachen (Syntaxdiagramme, EBNF, Fixpunktsemantik)
- Programmieren in *C* – Arrays, Pointer, Listen, Bäume
- grundlegende Algorithmen in der Informatik
  - Sortieren mit Quicksort und Heapsort
  - Suchen in Texten (KMP-Algorithmus)

Was heute geschieht ...

- Wiederholung: Suche mit dem KMP-Algorithmus
- Fehlerkorrektur mit der Levenshtein-Distanz
- Balancieren von Bäumen (AVL-Bäume)

KMP-Algorithmus

KMP-ALGORITHMUS — DIE ZWEI-FINGER-METHODE

Die Methode beruht auf der Gleichung

$$\begin{aligned}&\texttt{Tab[i]}\\&= \max\{-1\} \cup \{m \mid 0 \leq m \leq i-1 \wedge b_0 \ldots b_{m-i} = b_{i-m} \wedge b_{i-1} \wedge b_m \neq b_j\}\end{aligned} \quad (\star)$$

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 $i-1$, $i-2$, …), 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

Teil (a) Pattern: `abbabbaa`

AUFGABE 1

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

AUFGABE 1

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					c
Tabelle	-1	?	?	0	?	3

AUFGABE 1

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`

Levenshtein-Distanz

LEVENSHTEIN-DISTANZ

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

LEVENSHTEIN-DISTANZ

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

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

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

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

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

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

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 9.4

$$
\begin{array}{c|ccccccccccccccc}
d(j,i) & & & \textbf{D} & & \textbf{i} & & \textbf{s} & & \textbf{t} & & \textbf{a} & & \textbf{n} & & \textbf{z} \\
\hline
 & 0 & \rightarrow & 1 & \rightarrow & 2 & \rightarrow & 3 & \rightarrow & 4 & \rightarrow & 5 & \rightarrow & 6 & \rightarrow & 7 \\
 & \downarrow & \searrow & & & & & & & & & & & & & \\
\textbf{D} & 1 & & 0 & \rightarrow & 1 & \rightarrow & 2 & \rightarrow & 3 & \rightarrow & 4 & \rightarrow & 5 & \rightarrow & 6 \\
 & \downarrow & & \downarrow & \searrow & & & & & & & & & & & \\
\textbf{i} & 2 & & 1 & & 0 & \rightarrow & 1 & \rightarrow & 2 & \rightarrow & 3 & \rightarrow & 4 & \rightarrow & 5 \\
 & \downarrow & & \downarrow & & \downarrow & \searrow & & \searrow & & \searrow & & \searrow & & & \\
\textbf{n} & 3 & & 2 & & 1 & & 1 & \rightarrow & 2 & \rightarrow & 3 & & 3 & \rightarrow & 4 \\
 & \downarrow & & \downarrow & & \downarrow & \searrow & & \searrow & & \searrow & & \searrow & \downarrow & \searrow & \\
\textbf{s} & 4 & & 3 & & 2 & & 1 & \rightarrow & 2 & \rightarrow & 3 & \rightarrow & 4 & & 4 \\
 & \downarrow & & \downarrow & & \downarrow & & \downarrow & \searrow & & & & & & & \\
\textbf{t} & 5 & & 4 & & 3 & & 2 & & 1 & \rightarrow & 2 & \rightarrow & 3 & \rightarrow & 4 \\
 & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \searrow & & & & & \\
\textbf{a} & 6 & & 5 & & 4 & & 3 & & 2 & & 1 & \rightarrow & 2 & \rightarrow & 3 \\
 & \downarrow & & \downarrow & & \downarrow & \searrow & \downarrow & & \downarrow & & \downarrow & \searrow & & \searrow & \\
\textbf{s} & 7 & & 6 & & 5 & & 4 & & 3 & & 2 & & 2 & \rightarrow & 3
\end{array}
$$

AUFGABE 9.4

$d(j,i)$			**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

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

$d(j,i)$			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 9.4

$d(j,i)$		**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 9.4

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 2

$d(j,i)$			**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

AUFGABE 2

$$
\begin{array}{c|ccccccccccccccc}
d(j,i) & & & \mathbf{s} & & \mathbf{c} & & \mathbf{h} & & \mathbf{ü} & & \mathbf{r} & & \mathbf{z} & & \mathbf{e} \\
\hline
 & 0 & \rightarrow & 1 & \rightarrow & 2 & \rightarrow & 3 & \rightarrow & 4 & \rightarrow & 5 & \rightarrow & 6 & \rightarrow & 7 \\
 & \downarrow & \searrow & & \searrow & & \searrow & & \searrow & & \searrow & & \searrow & & \searrow & \\
\mathbf{b} & 1 & & 1 & \rightarrow & 2 & \rightarrow & 3 & \rightarrow & 4 & \rightarrow & 5 & \rightarrow & 6 & \rightarrow & 7 \\
 & \downarrow & \searrow & \downarrow & \searrow & & \searrow & & \searrow & & & & & & & \\
\mathbf{ü} & 2 & & 2 & & 2 & \rightarrow & 3 & & 3 & \rightarrow & 4 & \rightarrow & 5 & \rightarrow & 6 \\
 & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & & \searrow & \downarrow & \searrow & & & & & \\
\mathbf{r} & 3 & & 3 & & 3 & & 3 & \rightarrow & 4 & & 3 & \rightarrow & 4 & \rightarrow & 5 \\
 & \downarrow & \searrow & & \searrow & \downarrow & \searrow & \downarrow & \searrow & & & \downarrow & \searrow & & \searrow & \\
\mathbf{s} & 4 & & 3 & \rightarrow & 4 & & 4 & & 4 & & 4 & & 4 & \rightarrow & 5 \\
 & \downarrow & & \downarrow & \searrow & & \searrow & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & \\
\mathbf{t} & 5 & & 4 & & 4 & \rightarrow & 5 & & 5 & & 5 & & 5 & & 5 \\
 & \downarrow & & \downarrow & \searrow & \downarrow & \searrow & & \searrow & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & \\
\mathbf{e} & 6 & & 5 & & 5 & & 5 & \rightarrow & 6 & & 6 & & 6 & & 5
\end{array}
$$

$d(\text{bürste}, \text{schürze}) = 5$

AUFGABE 2

$d(j,i)$			**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

$d(\text{bürste}, \text{schürze}) = 5$

AUFGABE 2

$d(j,i)$		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

$d(\text{bürste}, \text{schürze}) = 5$

AUFGABE 2

$d(j,i)$		**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 2 — TEIL (B)

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

$$
\begin{array}{l|ccccccc}
d(j,i) & & & \mathbf{s} & & \mathbf{c} & & \mathbf{h} \\
\hline
 & 0 & \rightarrow & 1 & \rightarrow & 2 & \rightarrow & 3 \\
 & \downarrow & \searrow & & \searrow & & \searrow & \\
\mathbf{b} & 1 & & 1 & \rightarrow & 2 & \rightarrow & 3 \\
 & \downarrow & \searrow & \downarrow & \searrow & & \searrow & \\
\mathbf{ü} & 2 & & 2 & & 2 & \rightarrow & 3 \\
 & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & \\
\mathbf{r} & 3 & & 3 & & 3 & & 3 \\
 & \downarrow & \searrow & & \searrow & \downarrow & \searrow & \downarrow \\
\mathbf{s} & 4 & & 3 & \rightarrow & 4 & & 4 \\
 & \downarrow & & \downarrow & \searrow & & \searrow & \downarrow \\
\mathbf{t} & 5 & & 4 & & 4 & \rightarrow & 5
\end{array}
$$

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 2 — TEIL (B)

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

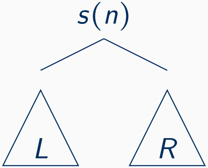
b	ü	r	s	t		b	ü	r	s	t
|	|	|	|	|		|	|	|	|	|
s	c	h	*	*		*	*	s	c	h
s	*s*	*s*	*d*	*d*		*d*	*d*	*s*	*s*	*s*

AVL-Bäume

Wir betrachten einen Baum t und bezeichnen die *Schlüssel* an den Knoten n mit $s(n)$.

Wir betrachten einen Baum t und bezeichnen die *Schlüssel* an den Knoten n mit $s(n)$.

Suchbaum:

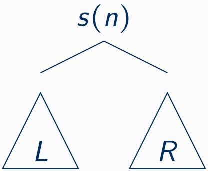


mit $L < s(n) < R$

AVL-BÄUME

Wir betrachten einen Baum t und bezeichnen die *Schlüssel* an den Knoten n mit $s(n)$.

Suchbaum:



mit $L < s(n) < R$

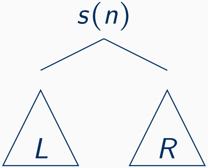
Die *Höhe* des Baumes bezeichnen wir mit $h(t)$. Wir ordnen jedem Knoten n einen *Balancefaktor* $b(n)$ zu:

$$b(n) := h(R) - h(L)$$

AVL-BÄUME

Wir betrachten einen Baum t und bezeichnen die *Schlüssel* an den Knoten n mit $s(n)$.

Suchbaum:



mit $L < s(n) < R$

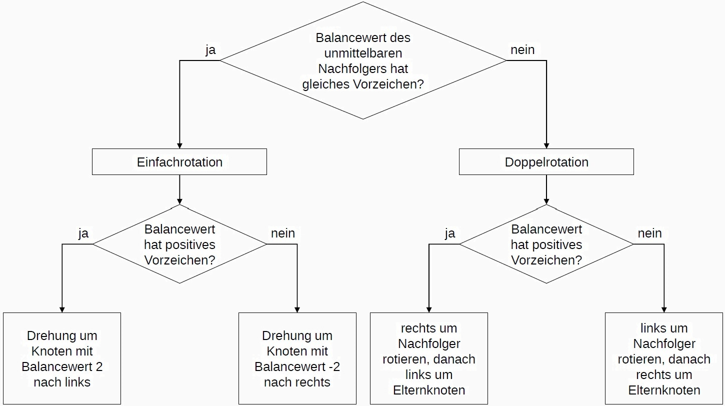
Die *Höhe* des Baumes bezeichnen wir mit $h(t)$. Wir ordnen jedem Knoten n einen *Balancefaktor* $b(n)$ zu:

$$b(n) := h(R) - h(L)$$

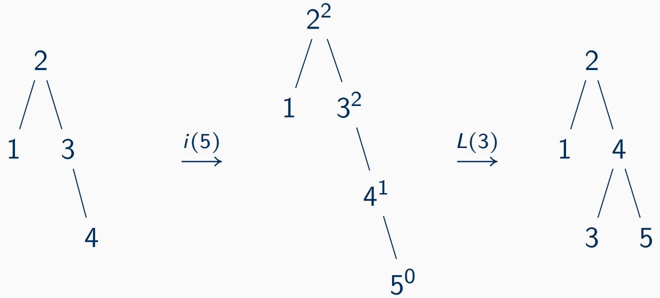
AVL-Baum: Suchbaum mit $b(n) \in \{-1, 0, 1\}$

BALANCIEREN

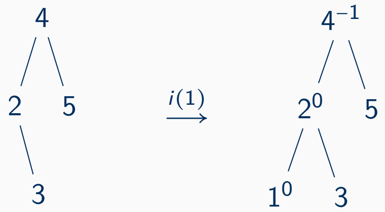
- Einfügen eines neuen Schlüssels s
- Berechne Balancefaktoren auf dem Pfad von s zur Wurzel bis zum ersten Auftreten von ± 2
- **Balancierungsalgorithmus**:



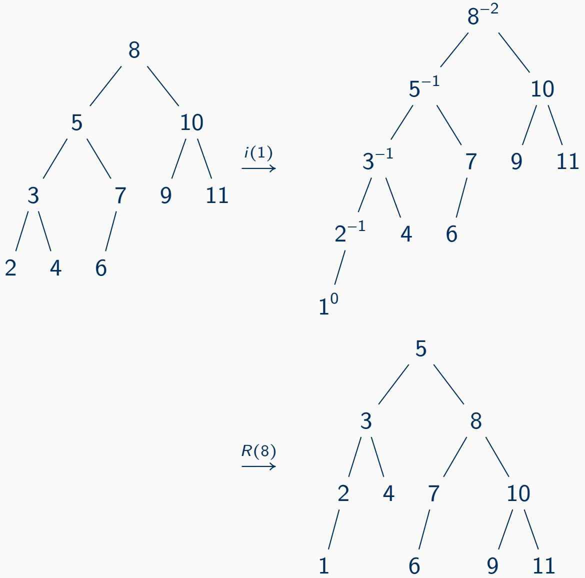
AUFGABE 3



AUFGABE 3



AUFGABE 3



AUFGABE 3