9. Übung AuD

Dominic Deckert

6. Januar 2017

Infos

VG Wort
Übung nächste Woche

Previously on ...

- Quicksort
- Heap-Sort
  - Heap-Eigenschaft
  - Sinkenlassen
  - Phase 1, 2

Aufgabe 1, 2

Ordne 9, 13, 7, 6, 10 mit Quicksort.
Ordne 2, 4, 17, 9, 13, 20, 12, 8, 5, 18 mit Heapsort.

KMP-Algorithmus

Algorithmus zum *pattern matching*
Annahme: Pattern-Vergleich schlägt an $i+1$-ter Stelle fehl:
$\rightarrow$ erste i Stellen gleich, $i+1$-te Stelle ungleich

KMP-Algorithmus

Algorithmus zum *pattern matching*

Annahme: Pattern-Vergleich schlägt an $i+1$-ter Stelle fehl:

$\rightarrow$ erste i Stellen gleich, $i+1$-te Stelle ungleich

	0	1	2	3	4	5	6	7		
Text:	...	a	a	b	a	a	b	a	a	...
Pattern:		a	a	b	a	a	c			
Match:		O	O	O	O	O	X			

Hinweis: Der Text ist nur zur Illustration und rein hypothetisch

KMP-Algorithmus

Algorithmus zum *pattern matching*

Annahme: Pattern-Vergleich schlägt an $i+1$-ter Stelle fehl:

$\rightarrow$ erste i Stellen gleich, $i+1$-te Stelle ungleich

	0	1	2	3	4	5	6	7		
Text:	...	a	a	b	a	a	b	a	a	...
Pattern:		a	a	b	a	a	c			
Match:		O	O	O	O	O	X			

Hinweis: Der Text ist nur zur Illustration und rein hypothetisch

Welche Pattern-Position muss an die Stelle des Mismatch geschoben werden?

Rechenwege

Intuition, Rechenregel aus dem Skript
Zwei-Finger-Methode:

1. Beginne links mit einem Abstand von 1
2. Solange der R nicht auf dem Mismatch liegt: Prüfe ob, $L = R$
   - Wenn ja: Verschiebe beide Finger
   - Wenn nein: Vergrößere Abstand und beginne von links
3. Wenn R auf dem Mismatch: Prüfe, ob $L \neq R$
   - Wenn ja: Verschiebe-Eintrag ist L
   - Wenn nein: Prüfe, ob Abstand maximal ist
     - Wenn ja: Verschiebe-Eintrag ist -1
     - Wenn nein: Vergrößere Abstand und beginne von links

Aufgabe 3 a

Pattern	a	a	b	a	a	a	c	a	a	b
	?	?	?	?	?	?	?	?	?	?

Pattern	c	b	?	?	?	a
	-1	0	-1	1	0	2

Aufgabe 3 a

Pattern	a	a	b	a	a	a	c	a	a	b
	-1	-1	1	-1	-1	2	2	-1	-1	0

Pattern	c	b	c	c	b	a
	-1	0	-1	1	0	2

AVL-Bäume

Erinnerung an C-Aufgabe: `isBalanced(tree t)`
Balancefaktor (*hier* BF): Höhe rechter TB - Höhe linker TB

AVL-Bäume

Erinnerung an C-Aufgabe: `isBalanced(tree t)`
Balancefaktor (*hier* BF): Höhe rechter TB - Höhe linker TB
Binärbaum ist balanciert gdw. TB sind balanciert & "fast" gleich hoch ($-1 \leq BF \leq 1$)

AVL-Bäume

Erinnerung an C-Aufgabe: `isBalanced(tree t)`
Balancefaktor (*hier* BF): Höhe rechter TB - Höhe linker TB
Binärbaum ist balanciert gdw. TB sind balanciert & "fast" gleich hoch ($-1 \leq BF \leq 1$)
AVL-Eigenschaft: balanciert & Werte im linken TB $<$ Wert der Wurzel $<$ Werte im rechten TB

AVL - Einfügen

1. Füge neuen Wert so als Blatt ein, dass Ordnung eingehalten wird
2. Berechne BF neu
3. Prüfe, ob BF 2 (bzw -2) auftritt
   - Ja: Prüfe, ob darunter BF mit entgegengesetztem Vorzeichen steht
     - Ja: Rotiere unten nach links (rechts), dann darüber nach rechts (links)
     - Nein: Rotiere unten nach links (rechts)
   - Nein: Fertig

Aufgabe 4

siehe Tafel