TU Kaiserslautern – FB Informatik
Dr. Patrick Michel

26.11.2019

Algorithmen und Datenstrukturen (WS 2019)

Aufgabenblatt 4

zu bearbeiten bis: 02.12.19 / 04.12.19

Aufgabe 4.1 (Sortieren am Beispiel - Theorie)
Sortieren Sie das folgende Array absteigend mit folgenden in der Vorlesung vorgestellten Verfahren:

2	1	7	8	4	3	9	6

- Insertionsort (Skizzieren Sie das Feld nach jedem Durchlauf der äußeren Schleife).
- Selectionsort (Skizzieren Sie das Feld nach jedem Durchlauf der äußeren Schleife).
- Optimierter Bubblesort (Skizzieren Sie das Feld nach jedem Durchlauf der äußeren Schleife).
- Mergesort (Skizzieren Sie das Feld nach jeder Misch-Operation).
- Quicksort (Skizzieren Sie das Feld nach jedem Aufruf von teile()).
- Radix Exchange Sort (Skizzieren Sie das Feld nach jedem Aufruf von teile()).

Aufgabe 4.2 (Merge- und QuickSort - Praxis)
Wir implementieren nun zwei effiziente Sortierverfahren.

a) Implementieren Sie (aufsteigenden) MergeSort in Java.
Hinweis: Für die Mischen-Operation legen Sie am einfachsten einen Zwischenspeicher an, den Sie dann im Anschluss an der richtigen Stelle in das zu sortierende Array kopieren. Dazu bietet `ArrayHelper` keine Methode an; schreiben Sie die Daten direkt ins Array (zum Beispiel über `helper.array()`).

b) Ist Ihre Implementierung stabil? Begründen Sie warum. Schlagen Sie eine Änderung vor, die das Verfahren instabil (falls es stabil ist) oder stabil (falls es instabil ist) macht.

c) Implementieren Sie (aufsteigenden) QuickSort in Java.

Achten Sie darauf, dass Ihre Verfahren auch für Elemente mit gleichem Schlüssel funktionieren. Ist die umgangssprachliche Beschreibung von QuickSort auf den Folien korrekt oder der Pseudocode? Begründen Sie warum!

`Materialien: https://softech.cs.uni-kl.de/homepage/de/teaching/WS19/algodat`