

6. Tutorenblatt zu Algorithmen I im SoSe 2017

<http://crypto.itl.kit.edu/index.php?id=799>
{bjoern.kaidel,sascha.witt}@kit.edu

Im Folgenden findet ihr einige unverbindliche Vorschläge zur Gestaltung des sechsten Tutoriums.

Das Sortieren mittels verschiedener Algorithmen sollte gut verstanden sein. Übt mit den Studenten daher das Sortieren mit Quicksort und geht insbesondere bei Quicksort nochmal auf den Rekursionsstapel sowie auf die Partitionierung ein. Außerdem wurde der Vergleich von Quicksort und Mergesort besprochen und sollte auf jeden Fall in den Tutorien angesprochen werden.

Darüber hinaus wurden Quickselect und ganzzahlige Sortierung behandelt. Dabei wurden Bucketsort und Radixsort besprochen, es ist daher sinnvoll die Funktionsweise jeweils kurz zu wiederholen und an Beispielen zu veranschaulichen. Außerdem wurde das Kapitel zu Prioritätslisten begonnen.

In der Übung werden Bucketsort für Zahlen im Intervall $[0, 1)$ sowie Prioritätslisten-Varianten behandelt.

Zusätzliche Aufgabenvorschläge zu Quicksort:

Aufgabe 1 (*Quicksort auf verketteten Listen*)

Vergleiche https://de.wikipedia.org/wiki/Quicksort#QuickSort_f.C3.BCr_verkettete_Listen.

- Wie kann man Quicksort auf verketteten Listen implementieren? Was ist die Schwierigkeit/der Unterschied hierbei?
- Was für ein Problem gibt es bei der Implementierung? Welches Sortierverfahren ist besser geeignet und warum?

Aufgabe 2 (*Iteratives Quicksort*)

Vergleiche https://de.wikipedia.org/wiki/Quicksort#Iteratives_Quicksort.

Wie kann Quicksort nicht-rekursiv implementiert werden? Wieviel zusätzlicher Speicherplatz wird dazu benötigt?

Zusätzliche Aufgaben zu ganzzahligem Sortieren:

Aufgabe 3 (*Ganzzahliges Sortieren*)

Wie kann man n ganze Zahlen im Intervall von 0 bis $n^3 - 1$ in Zeit $\mathcal{O}(n)$ sortieren?

Musterlösung:

Betrachte die Zahlen in Darstellung zu Basis n . Dann kann man die 3-stelligen Zahlen mit Radixsort in Zeit $\mathcal{O}(3(n + n)) = \mathcal{O}(n)$ sortieren.

Bemerkung: Wenn n eine Zweierpotenz ist ($n = 2^k$) und die Zahlen in Binärdarstellung gegeben, dann ist die i -te Stelle jeweils der i -te Block von k Bits, also direkt ablesbar.