

## Übungsserie 8

### Aufgabe 1: Merge-Sort

Es soll der *Merge-Sort* Algorithmus gemäss Folien-Skript implementiert werden.  
Wir setzen voraus, dass nur Sequenzen der Längen von ganzen 2er-Potenzen ( $2^n$ ) sortiert werden müssen.

Eine Ausgangslage befindet sich auf ILIAS.

Wir verwenden dazu nur *int*-Arrays in Java resp. *int*-Lists in Python wie bei der von Übung 1 *Insertion-Sort* und führen dazu zusätzliche Hilfs-Indexes ein um eine Sequenz zu simulieren (siehe `ai`, `bi`, `si` in `merge()`).

Hinweise für Java:

- Um Arrays zu kopieren kann z.B. `System.arraycopy()` verwendet werden.
- Die Java-Virtual-Machine soll mit den Parameter gemäss *Session-Log* zur Ausführung gebracht werden (`-Xint -Xms100M -Xmx100M`).

Nach der Implementierung sollen die gemessenen Laufzeiten beobachtet werden.

Entsprechen diese den Erwartungen?

Wie sehen diese Zeiten aus im Vergleich mit der *Insertion-Sort* Implementation von Übung 1, Aufgabe 4?

## Aufgabe 2: Quick-Sort

Gegeben ist folgende Sequenz: 7 3 6 4 2 1 5

Diese soll mittels *Quick-Sort* sortiert werden.

Als Pivot wird immer das mittlere Element genommen.

Array-Index des Pivot's =  $\text{AnzahlElementeDerSequenz} / 2$

Hinweis:

Ganzzahl-Division:  $3 / 2$  (resp.  $3 // 2$  in Python) = 1  $\rightarrow$  somit das zweite Element von drei  
oder:  $4 / 2$  (resp.  $4 // 2$  in Python) = 2  $\rightarrow$  somit das dritte Element von vier.

Zeichnen Sie in nachfolgendem Baum die Ausführung der Sortierung ein:

