

Projekt **MERGE****SORT** ("Sortieren durch Mischen")

Arbeitsaufträge:

1. Arbeite den Text **mergesort_16-04-2021.pdf** genau durch (Seite 1 bis Seite 2 oben, also ohne die Abschnitte zu Aufwandsbetrachtung und Speicherplatzbedarf).
Verdeutliche das Verfahren anhand eines Beispiels, um die 8-elementige Liste {7, 2, 5, 9, 3, 2, 6, 8} mit den Komponenten $a[0]$, $a[1]$, . . . , $a[7]$ zu sortieren.
2. Nimm den Python-Quelltext **SelectionSort_while-loop.py** als Vorlage, so daß insbesondere die Programmteile
 - Erstellung der Liste mit Zufallszahlen,
 - Ausgabe der Quellliste und der sortierten Liste,
 - Messung des Zeitbedarfs zum Sortierenerhalten bleiben.

Entferne diejenigen Programmzeilen, welche den Sortiervorgang gemäß dem Verfahren SelectionSort veranlassen.

Implementiere geeignet die in Python als Funktionen geschriebenen Subroutinen

merge(array, left, middle, right) (Quelltext: aus funktion_merge.py.txt)

und

sort(array, left, right) (Quelltext: aus mergesort_16-04-2021.pdf)

sowie den

Aufruf der Funktion **sort**,

der das Sortieren der aus den n Komponenten $a[0]$, $a[1]$, . . . , $a[n-1]$ bestehenden Liste **a** veranlaßt.

3. Vergleiche die Algorithmen **SelectionSort** und **MergeSort** hinsichtlich des Zeitbedarfs, um Listen mit jeweils 2000, 10000, 50000, 100000 Datenelementen zu sortieren.