**H5.1 Mergesort "almost *in-place*" ★★**

In dieser Aufgabe sollt ihr eine optimierte Variante von Mergesort implementieren, bei der ihr nur zwei Listen verwendet, zwischen denen ihr beim Mischen die Elemente jeweils hin und her kopiert (wie in **Präsenzaufgabe 4.2** diskutiert). Die grundlegende Idee sollte anhand des folgenden Beispiels noch einmal verdeutlicht werden:

| **Eingabeliste** | **Ausgabeliste** | **(bezieht sich jeweils auf das Mischen)** |
| --- | --- | --- |
| [4,6,8,2,5,3,1,7] | [0,0,0,0,0,0,0,0] | Startzustand |
| [4,6,8,2,5,3,1,7] | [4,6,2,8,3,5,1,7] | Mischen der Teillisten der Länge 1 |
| [4,6,2,8,3,5,1,7] | [4,6,8,2,5,3,1,7] | Vertauschen der Listen/ Ausgabeliste steht jetzt vorne |
| [4,6,2,8,3,5,1,7] | [2,4,6,8,1,3,5,7] | Mischen der Teillisten der Länge 2 / Mischen passiert in der Ausgabeliste |
| [2,4,6,8,1,3,5,7] | [4,6,2,8,3,5,1,7] | Vertauschen der Listen |
| [2,4,6,8,1,3,5,7] | [1,2,3,4,5,6,7,8] | Mischen der Teillisten der Länge 4 |

Die Ausgabeliste enthält am Ende das Ergebnis (falls am Ende nicht noch einmal vertauscht wird) und muss nur noch ggf. **in die ursprünglich zu sortierende Liste zurückkopiert werden.**

1. Realisiert zunächst eine neue Variante der merge-Funktion, welche nun mehr Parameter benötigt (mindestens Eingabeliste, Ausgabeliste und Länge der zu mergenden Teillisten in der Eingabeliste, ggf. auch deren Anfangspositionen in der Eingabeliste). Orientiert euch dazu am Beispiel.
2. Anschließend könnt ihr dann das eigentliche Mergesort realisieren. Hierbei sollen nur zwei Listen im Speicher gehalten (die übergebene Liste und eine weitere Liste gleicher Länge), die abwechselnd als Ein- und Ausgabeliste für den Merge-Schritt verwendet werden. 4 P.

**Hinweis:** Vermeidet es, beim Wechsel zwischen den Ein- und Ausgabeliste erneut alle Werte zu kopieren. Das Vertauschen der Referenzen ist aber in Ordnung, da eine Zuweisung einer Listenreferenz zu einer Variablen konstante Laufzeit hat.