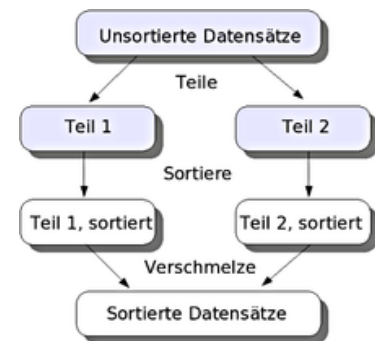


Praktikum 2: Merge-Sort

Ein bisher nicht betrachtetes Sortierverfahren ist der sog. Merge-Sort. Es handelt sich dabei wieder um einen "Teile-und-Herrsche"-Algorithmus. Die Grundidee ist dabei folgende:

- I. Um ein Array zu sortieren betrachtet man zunächst die Länge des Arrays. Hat es die Länge 1, ist es sortiert und wir sind fertig.
- II. Andernfalls teilen wir das Array in zwei möglichst gleich große Teilarrays (links und rechts) und rufen das Sortierverfahren für jedes erneut rekursiv auf. Anschließend müssen die beiden sortierten Teilarrays zusammengeführt werden ("Merge").



Beim "Merge" ist es sinnvoll, mit Hilfsarrays zu arbeiten, die die sortierten Teilarrays aufnehmen, weil das Mischen innerhalb des zu sortierenden Arrays (in-place) komplex ist.

1. Implementieren Sie den beschriebenen Algorithmus und sortieren Sie die vorgegebenen Folgen vom letzten Praktikum (seq0.txt bis seq3.txt). Messen Sie die Laufzeit des Sortiervorgangs.
2. Welche Laufzeitkomplexität ist bei diesem Algorithmus zu erwarten? Deckt sich dies mit Ihrer Messung? Sind hier Unterschiede bei einer Worst Case/Best Case-Betrachtung zu erwarten?
3. Noch eine Frage ohne direkten Bezug zum Merge-Sort: Welche der folgenden Aussagen sind wahr?
 - a. $f(n) \in O(f(n))$
 - b. $f(n) \in O(g(n)) \Rightarrow g(n) \in O(f(n))$
 - c. $f(n) \in \theta(g(n)) \Leftrightarrow g(n) \in \theta(f(n))$