Many different authors have proposed algorithms and strategies for in-place merging and merge sorting. Chen’s algorithm for example is itself a simplification of an in-place merging algorithm presented by geffert et al (<https://ac.els-cdn.com/S0304397598001625/1-s2.0-S0304397598001625-main.pdf?_tid=2299aab2-0cfe-469f-bfa7-42728ee3a0df&acdnat=1555496116_c2838578ebd0a45a3fea4231f452215c>).

An early approach to in-place merging include the algorithm of Mannila and Ukkonen (<https://ac.els-cdn.com/0020019084901121/1-s2.0-0020019084901121-main.pdf?_tid=b0f42859-2d06-46e9-9119-44e407e8313a&acdnat=1555494252_532858c4d1809ceaa9502bb2f43ab370>) that makes use of partitioning the given lists into blocks, that are then rearranged to provide for easier merging.

Huang and Langston later provided a much simpler approach also relying on the rearrangement of blocks.  
  
A more recent algorithm is presented by (..) (<https://ac.els-cdn.com/S1877050910005478/1-s2.0-S1877050910005478-main.pdf?_tid=56b08abc-b374-40ae-8a5f-9c685460c588&acdnat=1555417962_82e0e3ee735f1ee3ad9033616e9c2db3>), relying on an efficient in-place algorithm for shuffling two lists and the corresponding inverse operation. The shuffled lists can be efficiently sorted by consecutively inverse-shuffling parts that will afterwards be in correct order.