

I chose to implement the merge sort function recursively, with a helper method that worked iteratively. The approach to work recursively seemed best because the merge sort consistently breaks the list in half and has two branches coming out of it, and that repeats until the lists are length 1. I thought the best way to implement this branch splitting was to use recursion. However for the function to remerge all of the lists into one list I thought would be easier to implement iteratively as both lists could be iterated through with a while loop.

I did not have any issues coming up with my solution. I first assembled the base cases, and with the correct base cases in place, working out the solution came quite quickly.

The big-O complexity is $n \log n$ as all merge sort algorithms are.