CS 471 Senior Capstone I, Fall 2025
Writing Activity #2

Instructions

We wish to revise the following text to make it more readable and professional, and to better communicate information.

Copy the text and edit it on your own machine. As for formatting changes and similar things, you may make these changes, or simply remember what changes you would make. Be prepared to read your edited text and describe any formatting changes you would make.

Text

So, Merge Sort. It's probably the most commonly used sort these days. I read that somewhere. People think it's Quicksort, but they're wrong. I read that somewhere, too. There are lots of cool things about Merge Sort, like the fact that it's the fastest known sort for linked lists. And it's even in-place on linked lists. Anyway, it comes to us from my homie John von Neumann in 1945. (Think about that! 1945!) John von Neumann was awesome. We ought to sing a song about him every morning. Maybe it could be to the tune of Handel's Hallelujah Chorus (Handel's my other homie). Johhhhn von Neumann! Johhhhn von Neumann. John von Neumann! John von Neumann! You get the idea.

So, how does it work? Well, you get a list to sort. First you check if it's small. If you have zero items or just one, then nothing to do. Otherwise, make two parts. Recurse, recurse. Then put the two parts together into one big sorted part. That's merging (so Merge Sort; makes sense, right? Right???). Then you're done.

Anyway, cool stuff about it. It's stable. It's log-linear time. (You can do the merge in linear time, so Master Theorem, figure, figure, figure, yup, log-linear time.) It works on just about anything: arrays, linked lists, whatever. But there's one sorta nasty thing about it, and it's that to do the merge on an array you need to make a big ol' buffer, so linear extra space. Well, nobody's perfect.