Sorting

> *...After hours of sifting through all this data, you wish there were ways to sort this data. That'd make proving River's innocence a lot easier. Oh wait... You can.*

What is Sorting

Sorting is a type of algorithm to sort, say, an array into ascending numbers. There are many types of sorting algorithms that range in complexity and uses. The can be recursive (self-calling) or iterative (looping) or anything in between.

Algorithms of Sorting

Some algorithms are better at sorting in some other condition than others. For example, imagine you are sorting a deck of cards that has been well shuffled. Two algorithms might be:

1. Compare every card and put them in the right place. (Iterative)
2. Shuffle the deck and if check if they are sorted. If not, do it again. (Bogo sort)
3. Take one card and have another person sort the deck, who will do the same thing. (Recursion)

A normal person might choose algorithm 1 or 3 to implement. Someone that feels lucky might choose the second... Algorithm. The preformance of the sort depends on luck and algorithm design. A almost-fully sorted list would benefit from the first algorithm because it would end up in less comparisons. The third may be better for frazzled and completely unsorted arrays. The second has the potential to completely sort the array with O(n) complexity, which is better than the other two algorithms, but most likely than not, will have a complexity of O(undefined), meaning LOLMUHEFFICIENCY.

Stable vs Unstable

If we were to sort objects, we might sort them by the person's specifications. If the sort algorithm has more than one possible outcome, like a dog named fido and woofy could appear in the order fido -> woofy or woofy -> fido, then the algorithm is considered **unstable**. If the algorithm can only have one outcome, then it is a **stable** algorithm.

Visualization

Youtube "sort algorithms." Trust me you won't regret it its fun! :^)