Sorts

Parameters:

- inplace
- stability-maintain relative order for ”equal” keys

Accolades:

- simplest: selection
- most efficient: mergeSort
- insertion - best n average n^2 worst n^2
- selection - best n^2 average n^2 worst n^2
- merge - best $nlogn$ average $nlogn$ worst $nlogn$
- heap - best $nlogn$ average $nlogn$ worst $nlogn$
- quick - best $nlogn$ average $nlogn$ worst n^2

Classifications:

- simple (insertion, selection)
- efficient (Merge, Heap, quick)
- bubble
- distribution (bucket)
- exchange sorts (bubble sort, quicksort)
- selection sorts (shaker sort, heapsort)

Insertionsort:

- take one element at a time insert into place in another list
- first element
- 2nd after or before first

- 3rd in between after or before

Selectionsort:

- look for smallest, place
- next smallest, place

Quicksort:

- pick an element, called a pivot, from the array.
- reorder the array so that all elements with values less than the pivot come before the pivot, all elements with values greater than the pivot come after it (equal values can go either way)
- recursively apply the above steps to the sub-array of elements with smaller values and separately to the sub-array of elements with greater values

Mergesort:

- divide into two lists
- recursively sort 2 lists
- merge two lists (add in least of both comparing them first)

Bucketsort:

- create buckets by range (0-9,10-19,20-29,etc)
- separate into different buckets
- sort buckets (using either bucketsort, quicksort, what-ever)

Heapsort:

- create heap of all elements
- keep removing max (min) from heap to place into sorted array

Bubblesort:

- iterate through array swapping consecutive elements if in the wrong order
- run until an interation with no swaps

Sources:

- `http://betterexplained.com/articles/sorting-algorithms/`
- `https://en.wikipedia.org/wiki/Sorting_algorithm`