Group 18 sorting algorithm for Fall 2015 CSci 3501.

Elena Machkasova

Place: 3 (3rd on larger data set, 2nd on the smaller one).

Overall description

Key idea: store every **unique** string only **once**, keep track of the number of copies.

1. Go through input array, create a `DataObject` for every unique string.
   - If a string has occurred before, the counter in its `DataObject` is incremented.
   - `HashMap` is used to keep track of prior seen strings.
2. The `ArrayList` of non-duplicated strings (as `DataObjects`) is sorted using the default TimSort.
3. The result is written out to the given array: each string is added as many times as its counter indicates.

Running time: **2** traversals (to create `DataObjects` and to write out the results) and one sorting is the order **n log n**.

Details

1. `DataObject` computes/stores the following information upon creation:
   - The string numeric value (as `int`, computed once, used for HashMap lookup as well)
   - The sum of first 4 digits mod 10.
   - The reference to the original string (strings are never copied!).
2. `DataObject` implements Comparable.

Extra memory: `HashMap<Integer,DataObject>`, `ArrayList<DataObject>` (starts off as 1/20 of the total data size).

Disclaimer: I don't know which of these things are really beneficial, didn't have a chance to play with it.

Possible improvements: Using **Quicksort** instead of TimSort (since all elements are unique).