Group 8: Sorting Competition

George and Owen

Dataset 1: 79 79 80, Dataset 2: 86 80 89, 3rd place overall

Algorithm Results:

Dataset 1: 79 79 80, **Dataset 2**: 86 80 89, 3rd place overall

Correctness Issues: None!

Algorithms Used: Bucketsort on hamming distance, Timsort and Insertion sort within buckets

Runtime: O(n + L)

Additional Memory/Data Structures: Used an Array of ArrayLists for buckets, and concatenated to an array

Code Explanation:

1) Create an array A of size L, and fill array with ArrayLists
2) Look at each string in the given list, and insert it into the ArrayList that matches its hamming distance
3) After we have our buckets, if the bucket has a size <= 1, ignore it, size <= 32 run insertion sort, size > 32, run native TimSort, with a faster comparison.
4) The comparison looks at the first character of the string, if there the same, look at the next, if they are different, break and return the difference of the first and next strings character to get a valid comparison
5) After sorting each bucket, use addAll on a new ArrayList on each bucket, then return ArrayList.toArray

Other Things Tried/Would Try:

Tried:

- Radix sort
- Counting sort
- Quicksort
- Mergesort

Would try:

- Convert to int and back on result