Max Pursche
CS124 Project 4

Data

The dataset used was a rated list of different quick-cook ramen brands from around the world. It was updated last as of 2017, it has 2,580 different types of ramen. I removed a field that was unnecessary which identified different top ten year placements however the top tens weren't complete sets of year and date.

The following 6 fields are in the order of the CSV:

- Number - Integer, tracks the row number of a ramen
- Brand - String, The brand of a ramen
- Variety - String, The flavor of a ramen
- Style - String, The type of instant ramen that it is; a cup, a pack, a tray, or a bowl.
- Country - String, holds the country that a ramen is from
- Rating - Float, holds the rating of a ramen

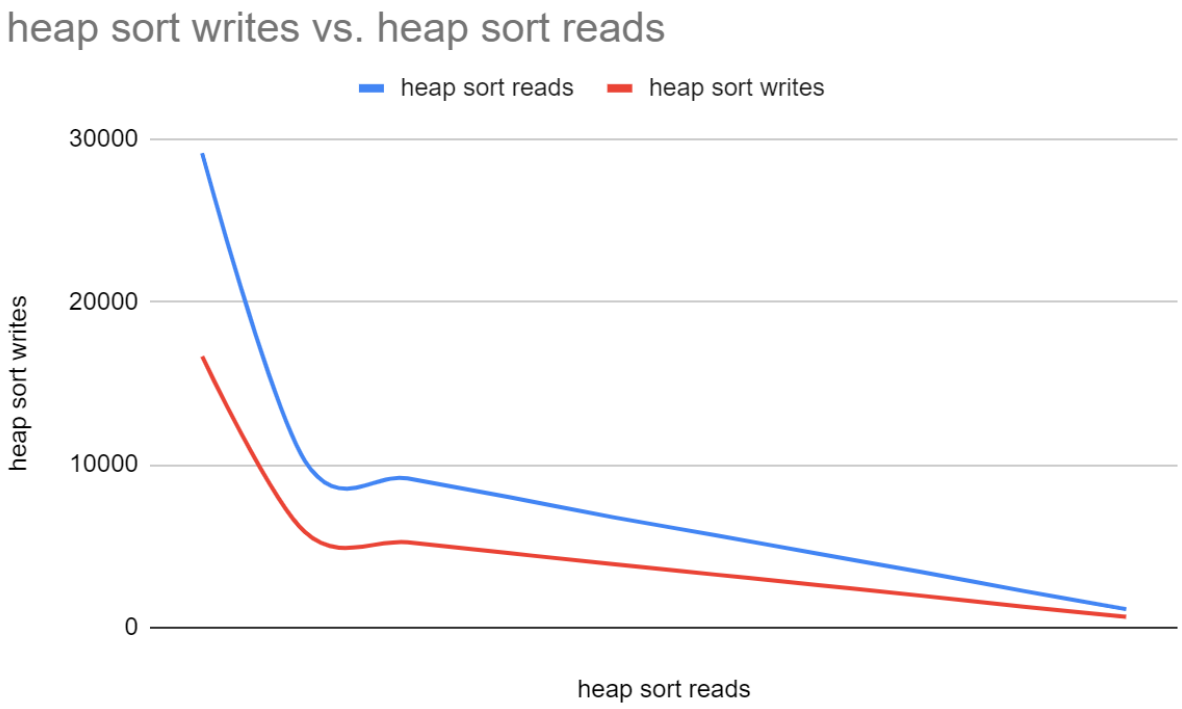
Entries are ordered from 2580 to 1.

Analysis

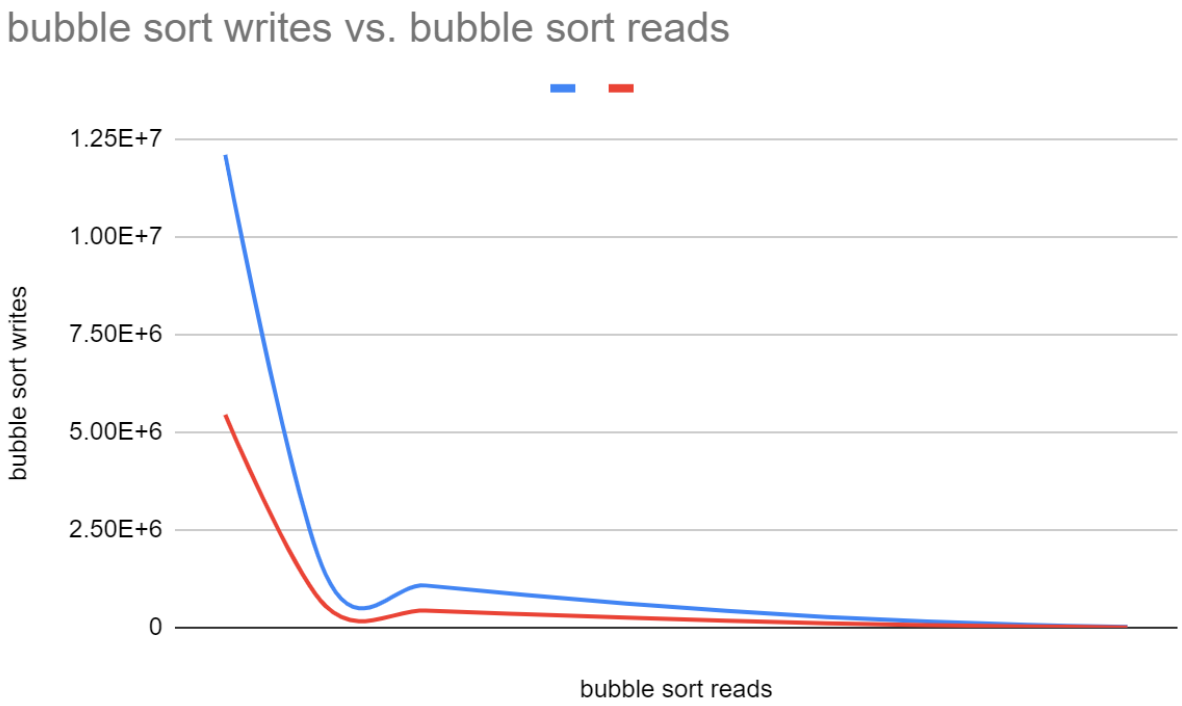
The reads and writes for all of the sorting algorithms seem to follow an exponential curve, the reads and writes after 1000+ vector objects seem to shoot up even more. Selection sort was the most efficient with my dataset, and twosort was the least efficient because it's technically two sorting algorithms. Bubble Sort also reads a lot of comparables, given the worst case complexity is n^2 it makes sense that as the vector gets larger the amount or reads and writes increases as well. Quick Sort was second most efficient, and this makes sense considering the nlog(n) complexity will be better suited to handle larger datasets while maintaining decent speeds at lower amounts.

two sort writes vs. two sort reads

quick sort writes vs. quick sort reads


heap sort writes vs. heap sort reads
heap sort reads
heap sort writes
30000
20000
10000
0
heap sort writes
heap sort reads


selection sort writes vs. selection sort reads
selection sort reads
selection sort writes
25000
20000
15000
10000
5000
0
selection sort writes
selection sort reads



If you need to sort a contacts list on a mobile app, which sorting algorithm(s) would you use and why?

Quick and Heap sort, they are NlogN and would be able to handle varying sizes of contact lists well.

What about if you need to sort a database of 20 million client files that are stored in a datacenter in the cloud?

Selection Sort for large datasets (i.e. databases)