Alex Moxon

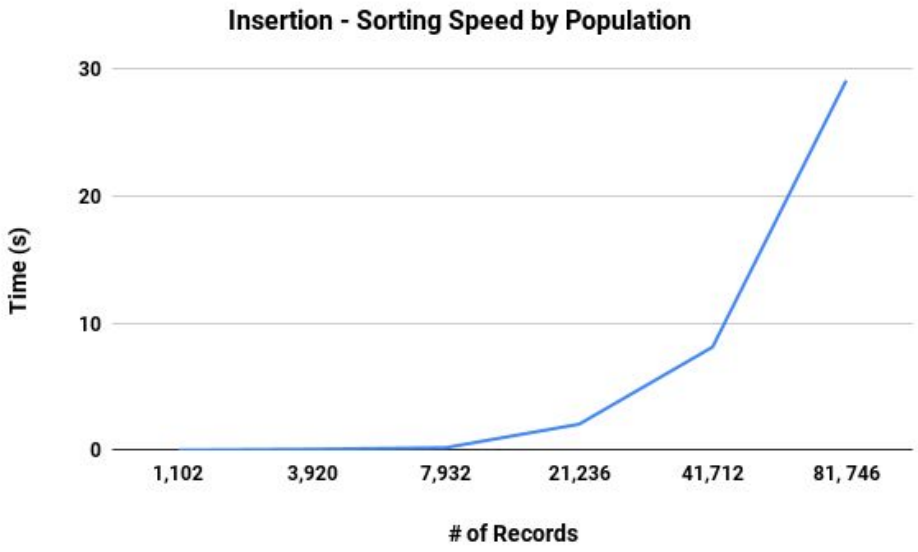
Prof. Challinger

CSCI 311-01

2/27/19

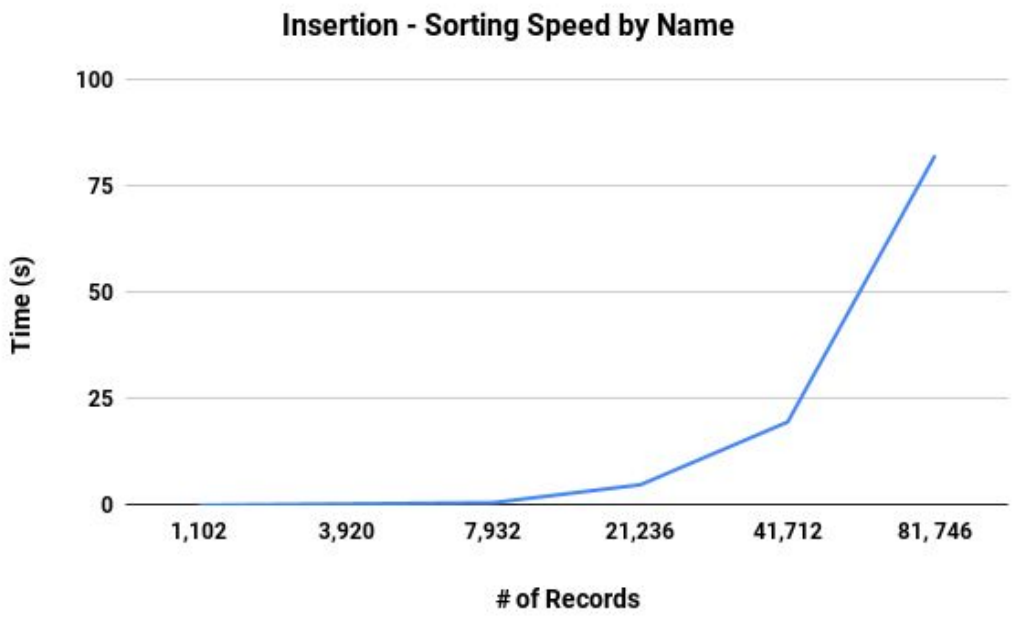
Project 2 - Sorting Report

Insertion Sort by Population:



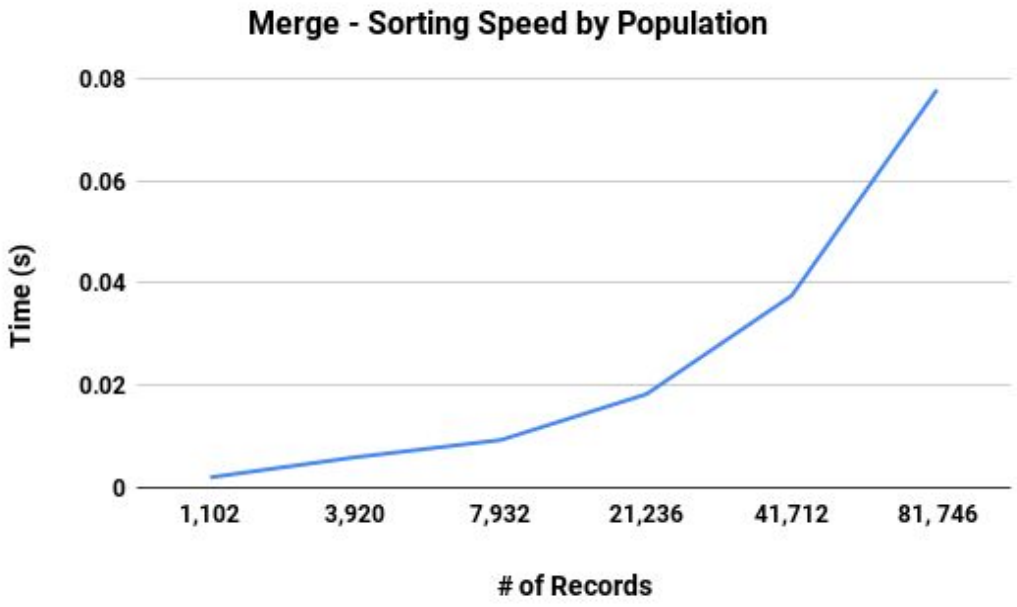
		Insertion Sort By Population (s)						
		1,102	3,920	7,932	21,236	41,712	81,746	# of Records
Runs	1	0.00837	0.07623	0.19977	2.18921	8.17093	33.72880	
	2	0.00831	0.08431	0.19899	2.21671	8.59101	30.88940	
	3	0.00616	0.08220	0.19967	2.04722	8.12672	29.11420	

Insertion Sort by Name:



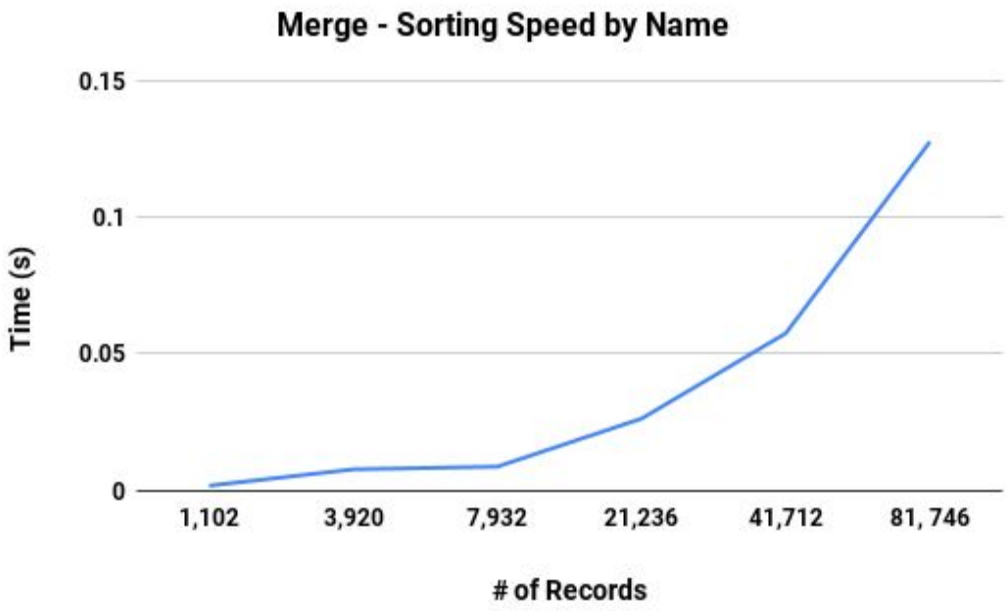
			Insertion Sort By Name (s)					
		1,102	3,920	7,932	21,236	41,712	81,746	# of Records
Runs	1	0.02676	0.32412	0.65329	5.27062	19.93860	83.47120	
	2	0.02631	0.34051	0.61731	4.82969	19.59260	82.24120	
	3	0.01980	0.33700	0.62070	4.82687	20.21860	84.24000	

Merge Sort by Population:



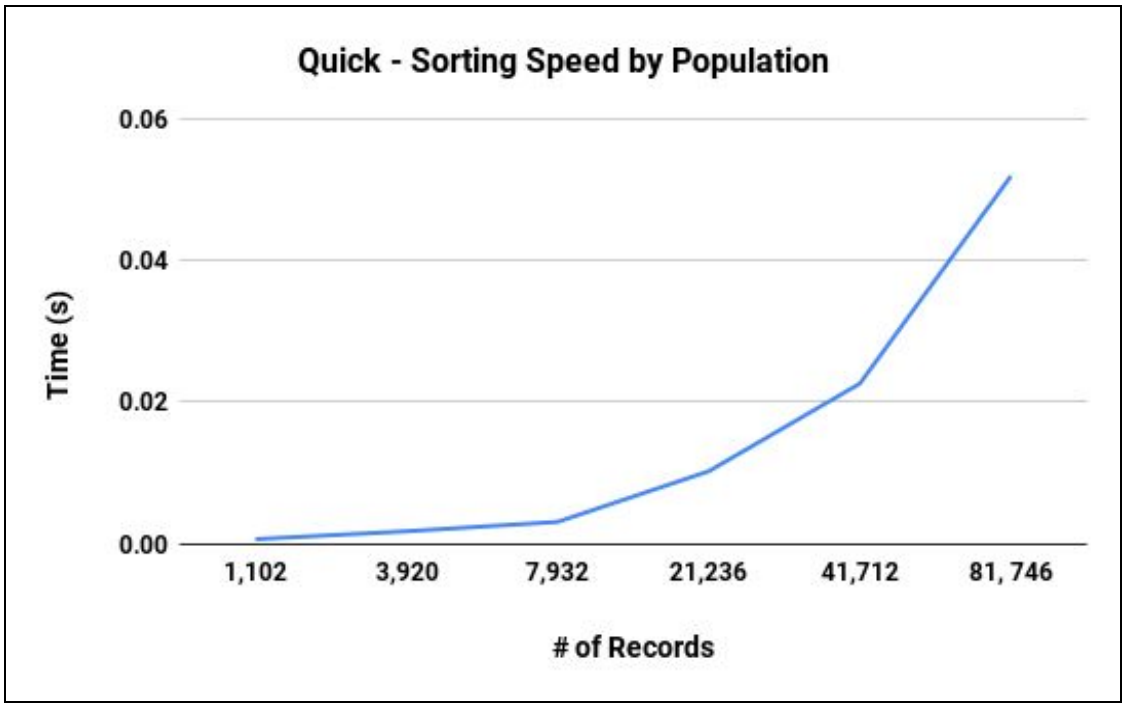
			Merge Sort By Population (s)					
		1,102	3,920	7,932	21,236	41,712	81,746	# of Records
Runs	1	0.001933	0.007342	0.006251	0.018228	0.040683	0.084187	
	2	0.001942	0.007338	0.006501	0.018287	0.037784	0.077902	
	3	0.001427	0.005873	0.006266	0.018676	0.037530	0.078131	

Merge Sort by Name:



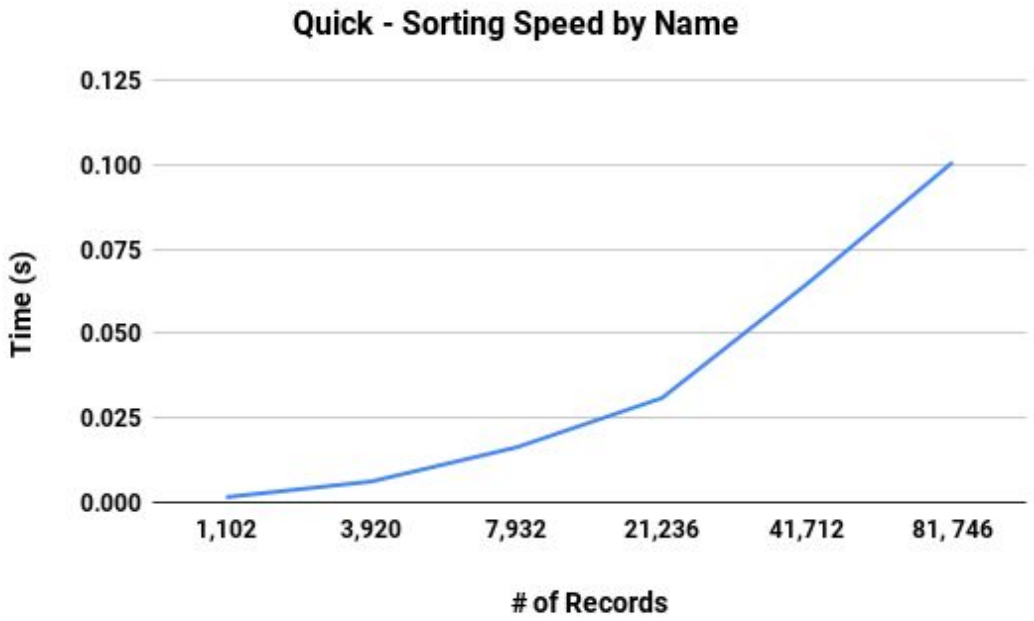
			Merge Sort By Name (s)					
		1,102	3,920	7,932	21,236	41,712	81,746	# of Records
Runs	1	0.002475	0.009639	0.008777	0.029564	0.058000	0.130103	
	2	0.002346	0.009722	0.008958	0.026707	0.057585	0.128111	
	3	0.001794	0.007780	0.008795	0.026358	0.058763	0.127764	

Quick Sort by Population:



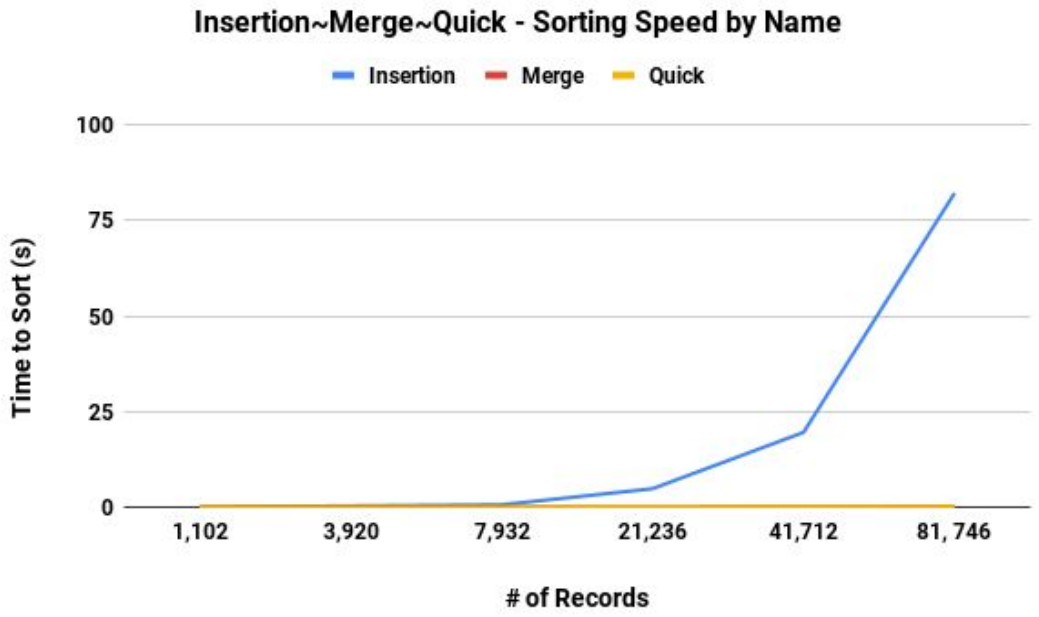
			QuickSort By Population (s)					
		1,102	3,920	7,932	21,236	41,712	81,746	# of Records
Runs	1	0.0009221	0.0037375	0.0033399	0.0107407	0.0232207	0.0526053	
	2	0.0008857	0.0036493	0.0031155	0.0103092	0.0226424	0.0519421	
	3	0.0006736	0.0027896	0.0033810	0.0102640	0.0227310	0.0547066	

Quick Sort by Name:



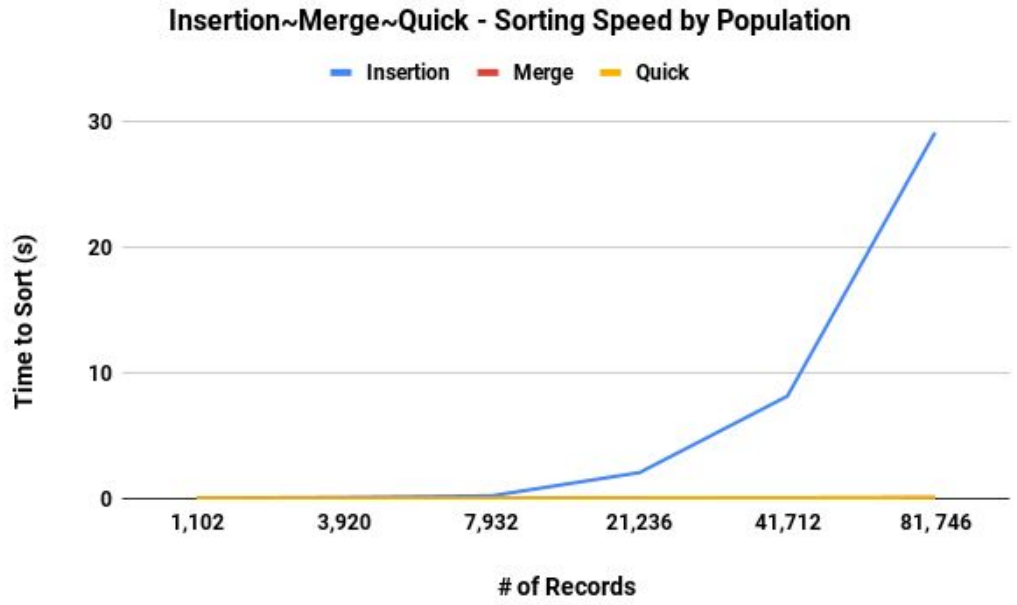
			QuickSort By Name (s)					
		1,102	3,920	7,932	21,236	41,712	81,746	# of Records
Runs	1	0.001497	0.006802	0.006703	0.022127	0.044720	0.102029	
	2	0.001568	0.006814	0.006131	0.020762	0.044840	0.100761	
	3	0.001325	0.005989	0.006704	0.020832	0.045237	0.105244	

Comparison by Name:



		All Sorts By Name (s)						
		1,102	3,920	7,932	21,236	41,712	81,746	# of Records
Sort	Insertion	0.01980	0.32412	0.61731	4.82687	19.59260	82.24120	
	Merge	0.001794	0.007780	0.008777	0.026358	0.057585	0.127764	
	Quick	0.001325	0.005989	0.006131	0.020762	0.044720	0.100761	

Comparison by Population:



		All Sorts By Population (s)						
		1,102	3,920	7,932	21,236	41,712	81,746	# of Records
Sort	Insertion	0.00616	0.07623	0.19899	2.04722	8.12672	29.11420	
	Merge	0.001427	0.005873	0.006251	0.018228	0.037530	0.077902	
	Quick	0.0006736	0.0027896	0.0031155	0.0102640	0.0226424	0.0519421	

Conclusion:

After multiple runs and from the data I've collected; Quick sort consistently performed the most optimal of the sorting functions for both population and name type. Insertion sort consistently performed the worst, becoming considerably less efficient the more records to be sorted. Merge sort performed really well when compared to Insertion sort but averaged about a 15~20% slower sorting speed when compared to Quick sort. Overall; Quick sort is the best general-purpose sorting function of the three we tested, even when accounting for number of records and for population and name type.