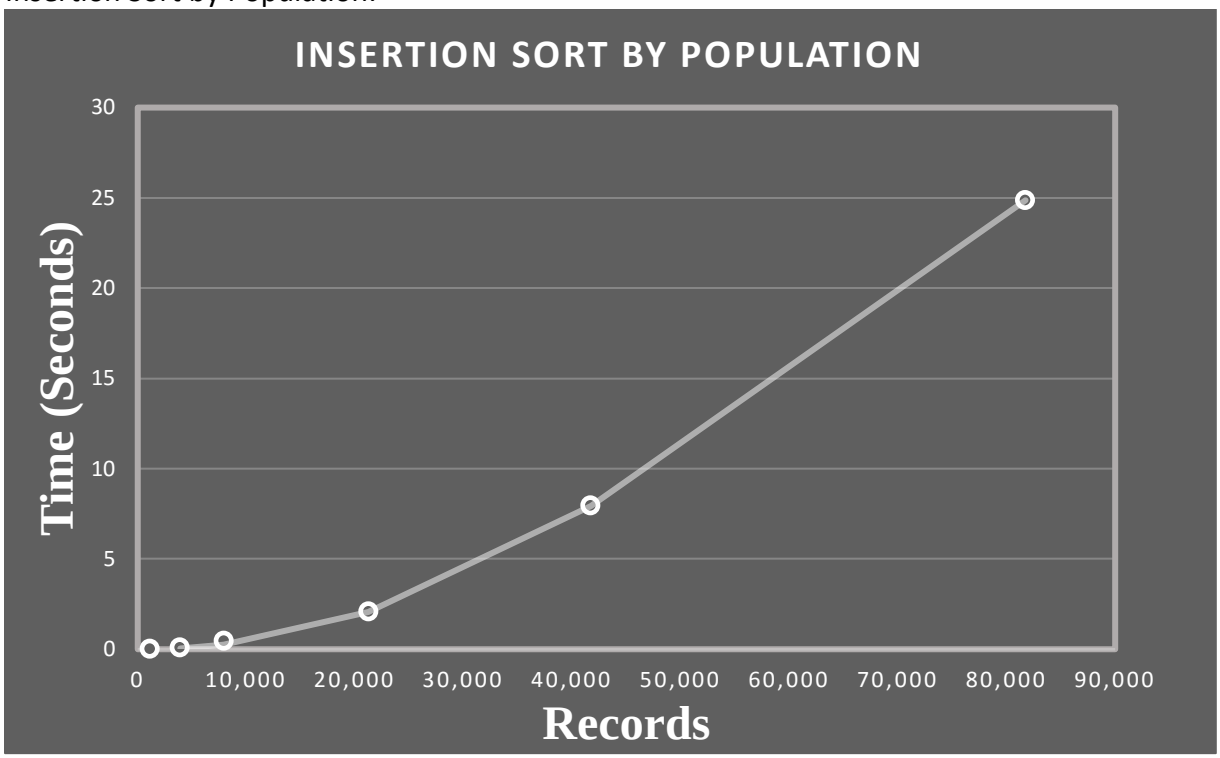
Insertion Sort, Merge Sort and Quick Sort

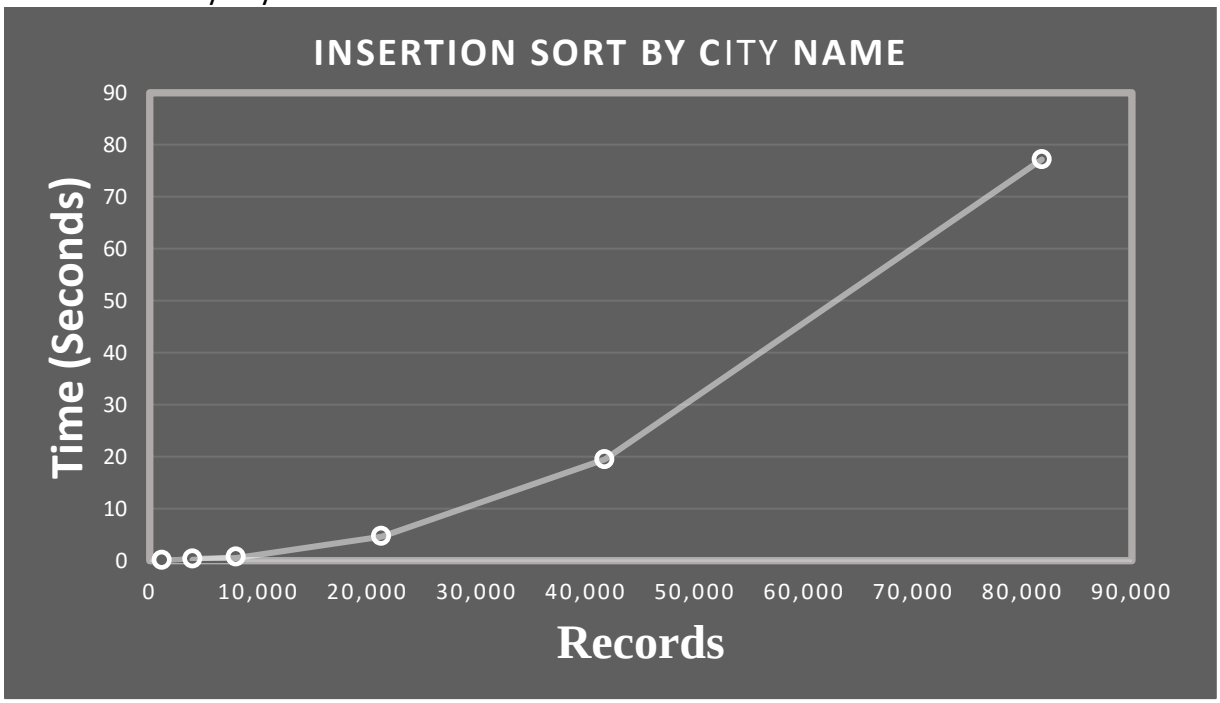
INSERTION SORT

Insertion Sort by Population:



Records:	Time Taken		
	Run 1	**Run 2**	**Run 3**
CENSUS2010POP-Alabama-Alabama	0.00770968	0.00782256	0.0082208
CENSUS2010POP-Alabama-California	0.0769514	0.0779004	0.0774381
CENSUS2010POP-Alabama-Idaho	0.463939	0.449395	0.451387
CENSUS2010POP-Alabama-Iowa	2.0739	2.15016	2.08483
CENSUS2010POP-Alabama-Missouri	10.7985	7.93521	8.51415
CENSUS2010POP	32.6673	31.3907	24.8655

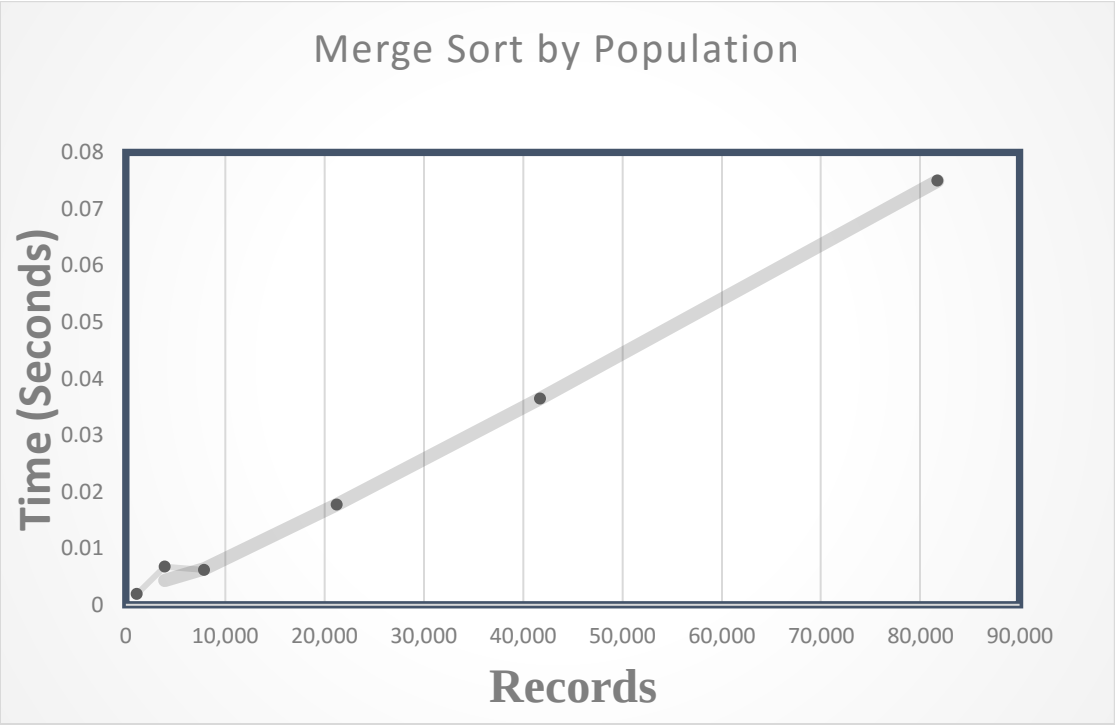
Insertion Sort by city name:



Records	Time Taken		
	Run 1	**Run 2**	**Run 3**
CENSUS2010POP-Alabama-Alabama	0.0252066	0.0262206	0.0253316
CENSUS2010POP-Alabama-California	0.331377	0.331611	0.328768
CENSUS2010POP-Alabama-Idaho	0.770242	0.678051	0.857111
CENSUS2010POP-Alabama-Iowa	4.80182	4.85075	4.62911
CENSUS2010POP-Alabama-Missouri	19.4855	21.259	20.6865
CENSUS2010POP	90.3489	87.5299	77.1736

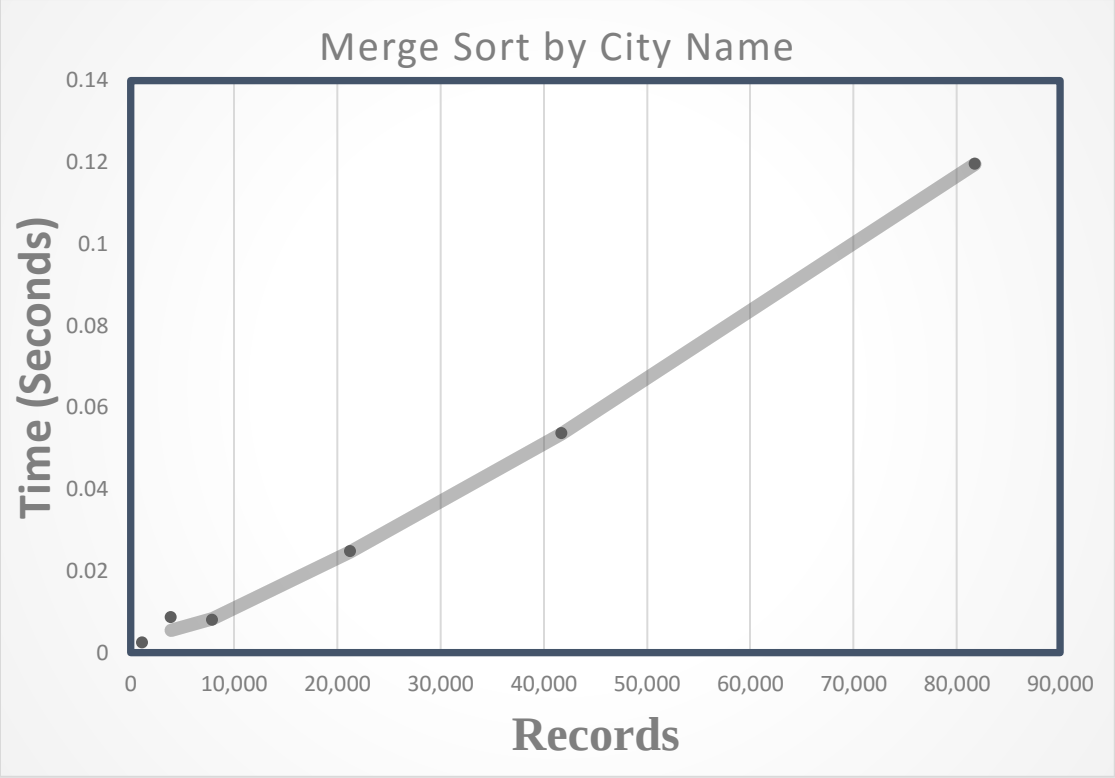
MERGE SORT

Merge Sort by Population:



Records	Time Taken		
	Run 1	**Run 2**	**Run 3**
CENSUS2010POP-Alabama-Alabama	0.00187183	0.00187002	0.00189912
CENSUS2010POP-Alabama-California	0.00726006	0.00673814	0.00711338
CENSUS2010POP-Alabama-Idaho	0.00670191	0.00619976	0.00610081
CENSUS2010POP-Alabama-Iowa	0.0178595	0.0180547	0.0176568
CENSUS2010POP-Alabama-Missouri	0.040134	0.0364116	0.0371969
CENSUS2010POP	0.0791932	0.191494	0.0749506

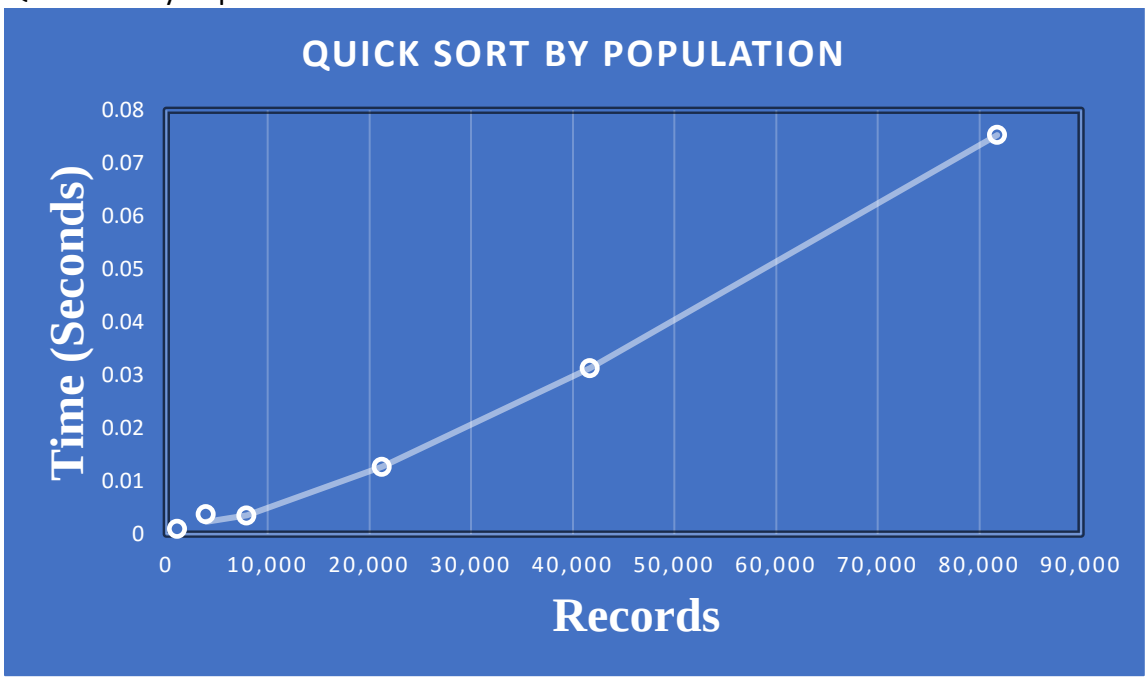
Merge Sort by City name:



Records	Time Taken		
	Run 1	**Run 2**	**Run 3**
CENSUS2010POP-Alabama-Alabama	0.00228446	0.00244779	0.00229326
CENSUS2010POP-Alabama-California	0.0106659	0.00862866	0.00915828
CENSUS2010POP-Alabama-Idaho	0.00888358	0.00795965	0.00809769
CENSUS2010POP-Alabama-Iowa	0.0251244	0.0246966	0.0246424
CENSUS2010POP-Alabama-Missouri	0.0650045	0.0535856	0.0609227
CENSUS2010POP	0.129839	0.30281	0.119537

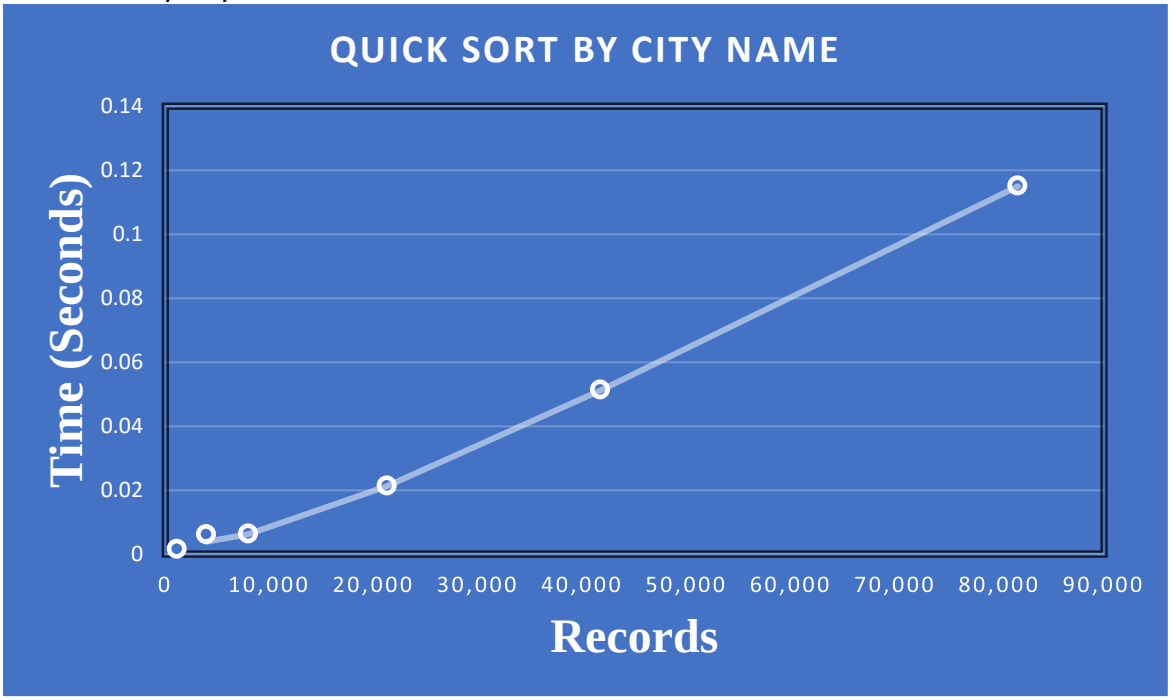
QUICK SORT

Quick Sort by Population:



Records:	Time Taken		
	Run 1	**Run 2**	**Run 3**
CENSUS2010POP-Alabama-Alabama	0.000962316	0.00113118	0.000947145
CENSUS2010POP-Alabama-California	0.00392337	0.00369725	0.00390775
CENSUS2010POP-Alabama-Idaho	0.00383212	0.00347187	0.00382385
CENSUS2010POP-Alabama-Iowa	0.0131858	0.0127349	0.0127323
CENSUS2010POP-Alabama-Missouri	0.0332137	0.0313711	0.0313101
CENSUS2010POP	0.0771374	0.195545	0.0753195

Quick Sort by City name:



Records	Time Taken		
	Run 1	**Run 2**	**Run 3**
CENSUS2010POP-Alabama-Alabama	0.00155317	0.00163593	0.0015885
CENSUS2010POP-Alabama-California	0.00666187	0.00614194	0.00698173
CENSUS2010POP-Alabama-Idaho	0.00690263	0.00633239	0.00714841
CENSUS2010POP-Alabama-Iowa	0.0249683	0.0222566	0.0212561
CENSUS2010POP-Alabama-Missouri	0.0589208	0.0530342	0.0512451
CENSUS2010POP	0.11298	0.172444	0.114965

Conclusion: After analyzing these three algorithms with three different runs for each one, I concluded that insertion sort is the has the lowest order of growth O(n^2) from all three, which makes it ineffective to sort large data. However, the data that was sorted was not in order, so insertion sort took a lot of time to sort it. Insertion Sort is better for small data. On the other hand, Merge Sort and Quick sort with the divide-and-conquer technique had an efficient time to sort these large data in the analysis. But merge Sort with its worst case and best case order of growth 0(n log n) performs a better sorting with large data, and as we can see in the graphs, it took almost the same time to sort the data that was given in the input file. For quick sort, it takes O(n^2) for worst case, and as we can see on the graphs, quick sort depends on the number of data that needs to be sorted. As can be seen merge sort is the most efficient algorithm when we try to sort large data.