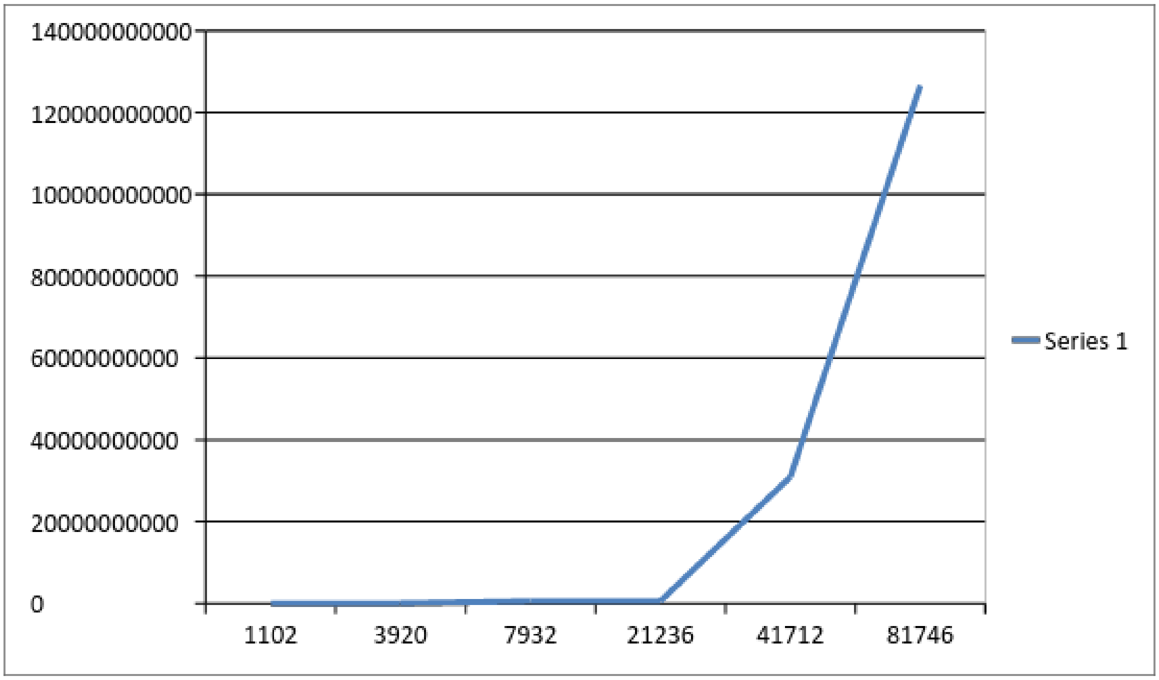
Sorting Algorithm Time Stamps

INSERTION SORT POPULATION

	CENSUS2010POP-Alabama-Alabama	CENSUS2010POP-Alabama-California	CENSUS2010POP-Alabama-Idaho	CENSUS2010POP-Alabama-Iowa	CENSUS2010POP-Alabama-Missouri	CENSUS2010POP
First	0:10466072	0:110453860	0:566090243	0:594259097	31:750583875	129:410087627
Second	0:10398107	0:102974050	0:563245029	0:565375287	31:917784286	131:100454231
Third	0:10627406	0:104663991	0:565343902	0:569290709	30:929789021	137:424744626
Fourth	0:10449404	0:103213895	0:568262708	0:567865733	33:504533425	126:601352119
Fifth	0:10440253	0:107751297	0:809589892	0:569756796	31:956804683	145:654982378
Smallest	0:10398107	0:102974050	0:563245029	0:565375287	30:929789021	126:601352119

X axis: Number of records present in census files

Y axis: time taken by CPU in nanoseconds

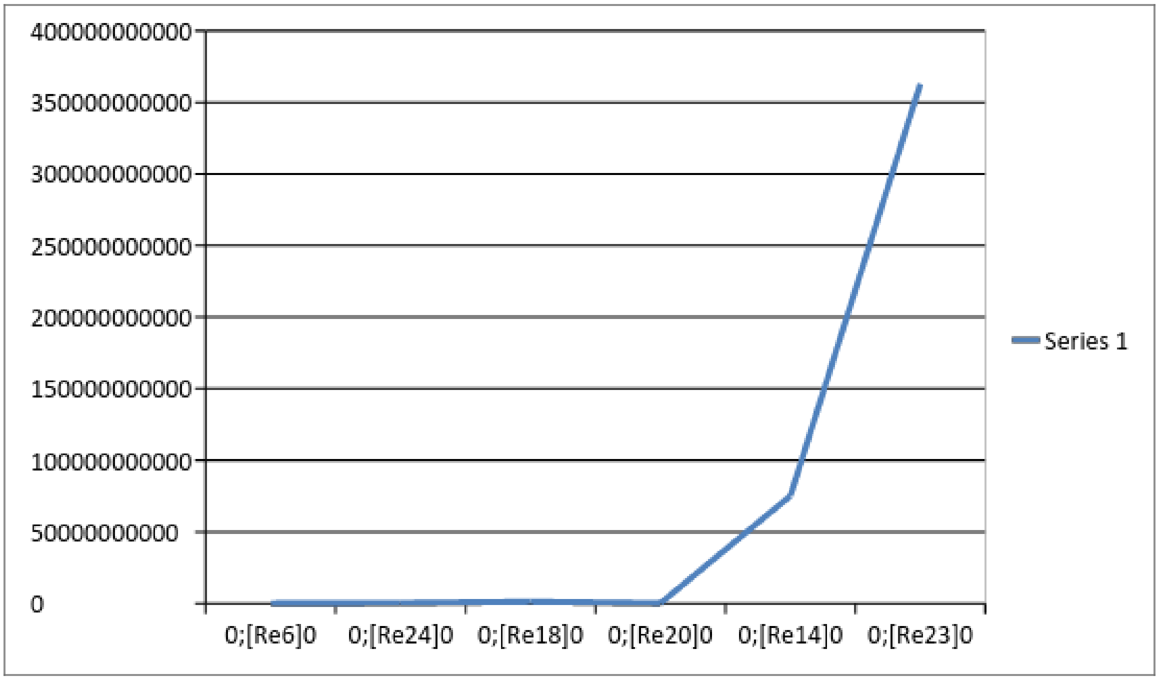


INSERTION SORT NAME

	CENSUS2010POP-Alabama-Alabama	CENSUS2010POP-Alabama-California	CENSUS2010POP-Alabama-Idaho	CENSUS2010POP-Alabama-Iowa	CENSUS2010POP-Alabama-Missouri	CENSUS2010POP
First	0:19857284	0:270910626	1:239041050	1:227872960	87:192431331	385:438678148
Second	0:19950467	0:268156101	1:229314127	1:368646932	75:126725175	362:693192351
Third	0:20023554	0:278444060	1:235734765	1:246329543	88:100450712	372:109244532
Fourth	0:19751095	0:270068581	1:240675173	1:243676094	90:804669423	368:857555826
Fifth	0:19811667	0:268935402	1:583077096	1:254970694	93:43898771	378:896623958
Smallest	0:19751095	0:268156101	1:229314127	1:227872960	75:126725175	362:693192351

X axis: Number of records present in census files

Y axis: time taken by CPU in nanoseconds

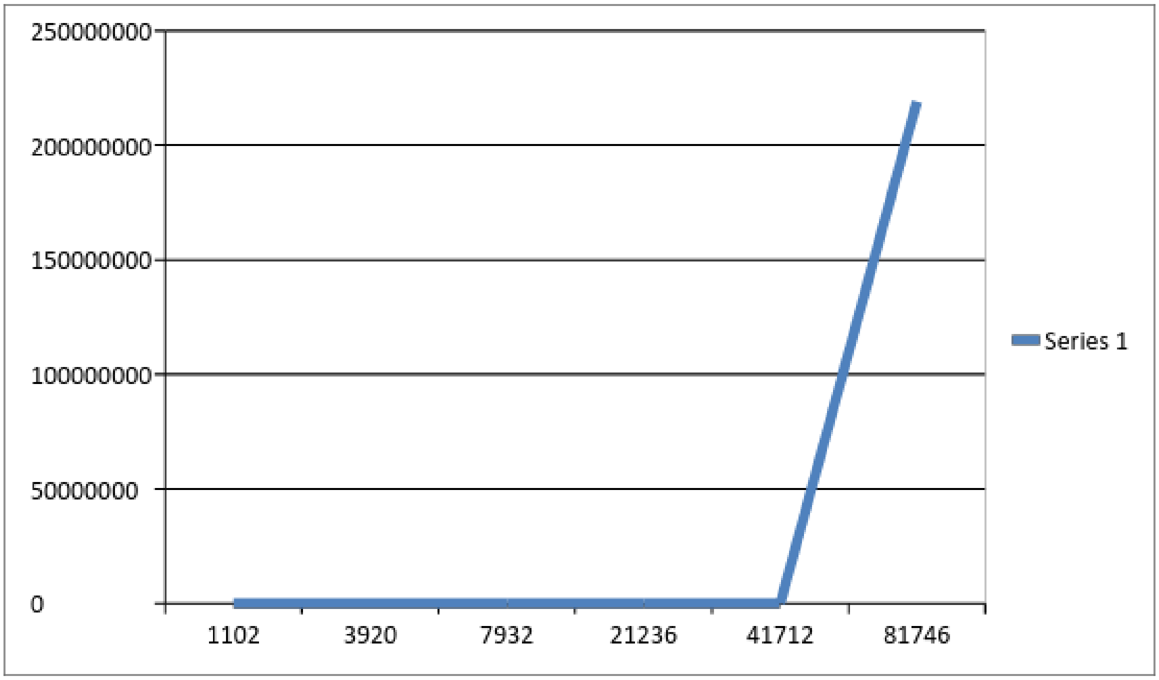


MERGE SORT POPULATION

	CENSUS2010POP-Alabama-Alabama	CENSUS2010POP-Alabama-California	CENSUS2010POP-Alabama-Idaho	CENSUS2010POP-Alabama-Iowa	CENSUS2010POP-Alabama-Missouri	CENSUS2010POP
First	0:2148563	0:7824126	0:16421879	0:16735928	0:113419313	0:228900358
Second	0:2147540	0:7951949	0:16773312	0:16356388	0:128792797	0:220641146
Third	0:2066622	0:7847723	0:17581273	0:17972630	0:109444355	0:219067653
Fourth	0:2057365	0:7853789	0:16814171	0:16747899	0:111660539	0:223455088
Fifth	0:2507818	0:7834105	0:25112440	0:16650211	0:109616719	0:289193359
Smallest	0:2057365	0:7824126	0:16421879	0:16356388	0:109444355	0:219067653

X axis: Number of records present in census files

Y axis: time taken by CPU in nanoseconds

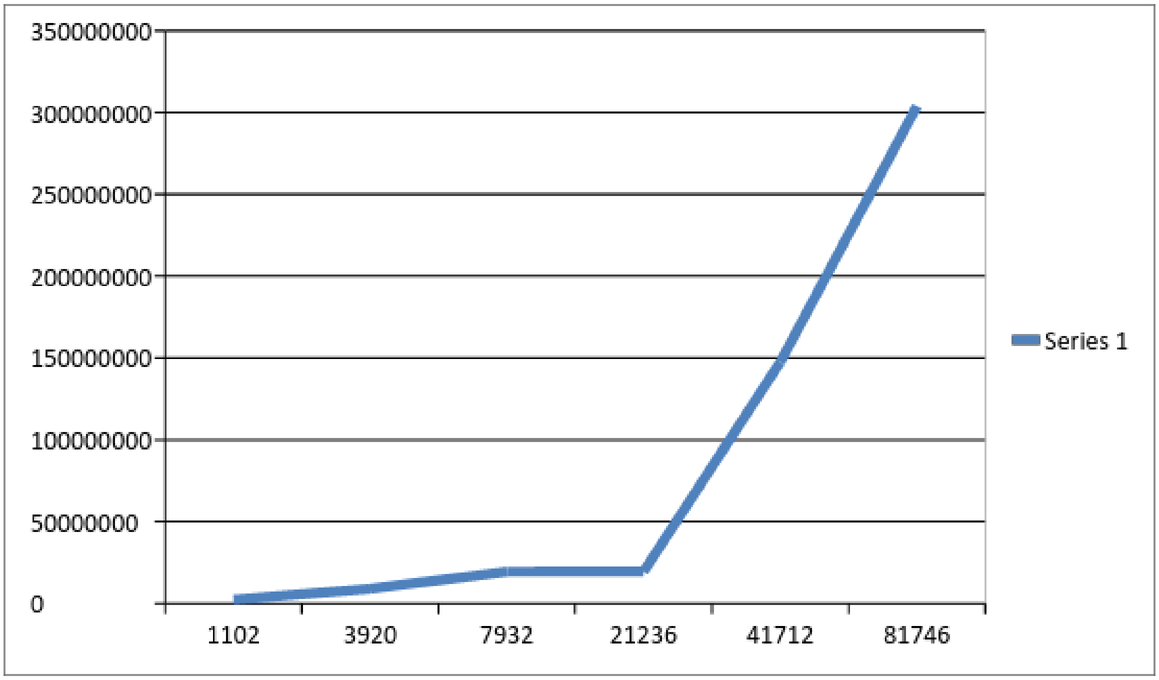


MERGE SORT NAME

	CENSUS2010POP-Alabama-Alabama	CENSUS2010POP-Alabama-California	CENSUS2010POP-Alabama-Idaho	CENSUS2010POP-Alabama-Iowa	CENSUS2010POP-Alabama-Missouri	CENSUS2010POP
First	0:2290507	0:9007542	0:19316099	0:19437558	0:154967532	0:356425385
Second	0:2286025	0:9316354	0:20074962	0:19410217	0:171535741	0:303890708
Third	0:2292971	0:9085969	0:20437282	0:20792317	0:147964036	0:304399366
Fourth	0:2290239	0:9013731	0:19929990	0:19784864	0:150289128	0:318332721
Fifth	0:2481334	0:9000437	0:23195481	0:19945550	0:147434695	0:434922983
Smallest	0:2286025	0:9000437	0:19316099	0:19410217	0:147434695	0:303890708

X axis: Number of records present in census files

Y axis: time taken by CPU in nanoseconds

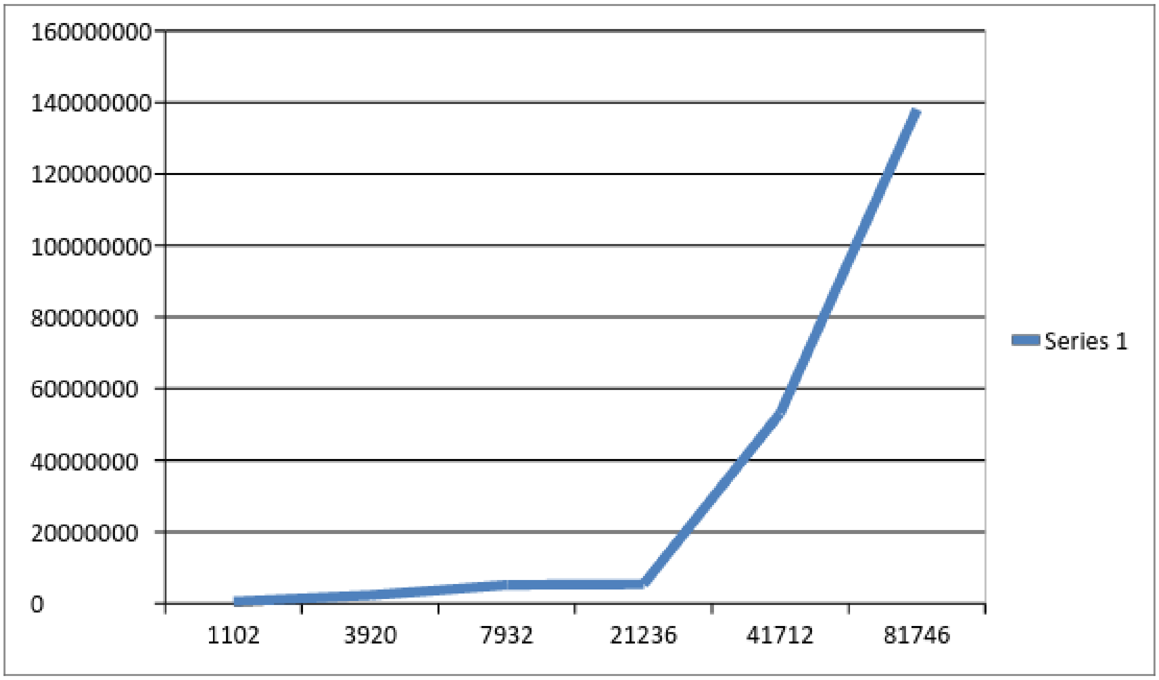


QUICK SORT POPULATION

	CENSUS2010POP-Alabama-Alabama	CENSUS2010POP-Alabama-California	CENSUS2010POP-Alabama-Idaho	CENSUS2010POP-Alabama-Iowa	CENSUS2010POP-Alabama-Missouri	CENSUS2010POP
First	0:602068	0:2525281	0:6280134	0:5486709	0:73766341	0:139444657
Second	0:530053	0:2332965	0:5812620	0:5357844	0:63800474	0:138003840
Third	0:527476	0:2343006	0:5162863	0:5734149	0:101079978	0:141060932
Fourth	0:574718	0:2392705	0:5534706	0:5624693	0:110966084	0:156162412
Fifth	0:531140	0:2258904	0:5714904	0:5748764	0:53160115	0:219071782
Smallest	0:527476	0:2258904	0:5162863	0:5357844	0:53160115	0:138003840

X axis: Number of records present in census files

Y axis: time taken by CPU in nanoseconds

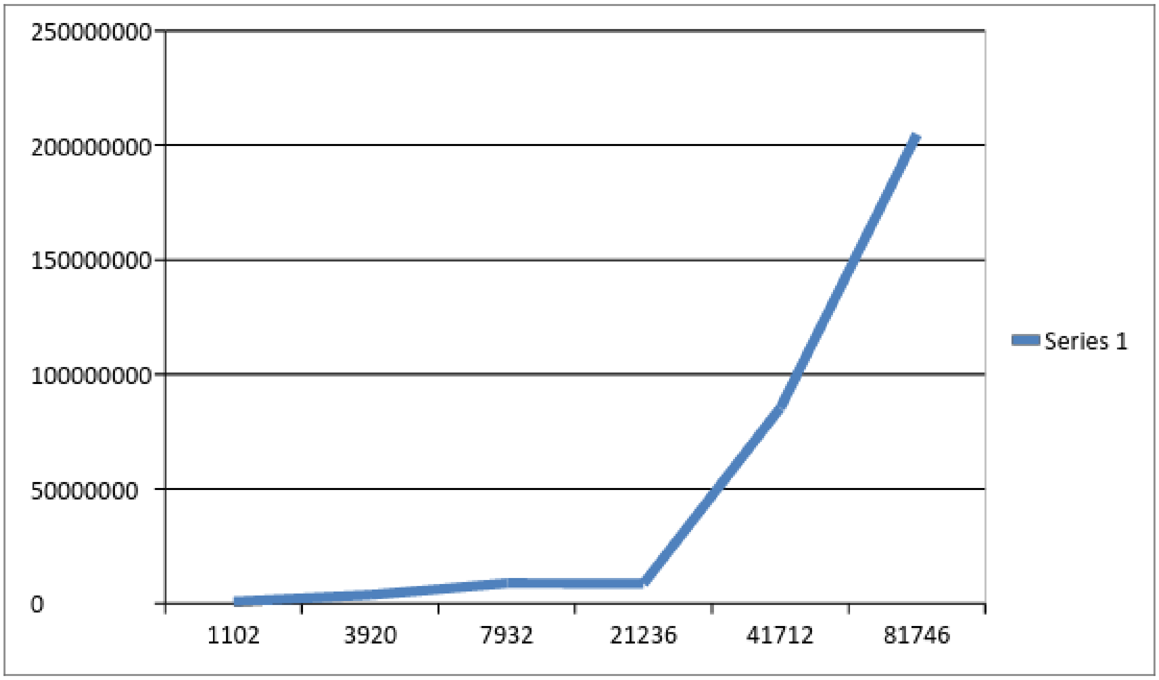


QUICK SORT NAME

	CENSUS2010POP-Alabama-Alabama	CENSUS2010POP-Alabama-California	CENSUS2010POP-Alabama-Idaho	CENSUS2010POP-Alabama-Iowa	CENSUS2010POP-Alabama-Missouri	CENSUS2010POP
First	0:809562	0:3675459	0:10614410	0:10007141	0:105274652	0:204739113
Second	0:829806	0:3783962	0:8801615	0:8672891	0:102499770	0:205139839
Third	0:833550	0:3857198	0:10563632	0:9451988	0:85296767	0:205001759
Fourth	0:827694	0:3772796	0:8916517	0:9118482	0:111160066	0:221819301
Fifth	0:844261	0:4026298	0:10638341	0:10040065	0:97677879	0:300096945
Smallest	0:809562	0:3675459	0:8801615	0:8672891	0:85296767	0:204739113

X axis: Number of records present in census files

Y axis: time taken by CPU in nanoseconds



Conclusion: Above six graphs are displaying number of records v/s time taken by CPU (in nanoseconds) to sort those records.

Conclusion is based on number of records present in a dataset or file, and type of the data to be sorted.

1. For small size of data:

Insertion sort algorithm is more efficient than merge sort and quick sort algorithms.

For example, in above graph, if we consider the smaller file of 1,102 records, time taken by CPU to sort those records using insertion sort is much less than other two sorts i.e. merge sort and quick sort, and quick sort takes maximum CPU time among three of the sorting algorithms.

2. For large size of data:

For large data to be sorted, quick sort algorithm works more efficiently than merge sort and insertion sort algorithms.

For example, in above graphs, if we consider the larger file such that a file of 81746 records, quick sort takes less time less CPU time to sort those records as compared to insertion sort and merge sort, and insertion sort takes maximum CPU time. So, we can conclude that if the dataset to be sorted is small then in that case, insertion sort is more time efficient than merge and quick sort. However, for sorting large dataset, quick sort is more time efficient than other two sorts, and insertion sort algorithm will not work efficiently.

3. For different type of data:

Within these three sorting algorithms, CPU time for sorting data changes, if there exists different type of data. Sorting time taken by CPU to sort data according to string(city) is greater than that of integer (population).

For an instance, in insertion sort time taken by CPU to sort data by string(city) takes more time than time taken by CPU to sort data by integer(population). For 7932 records, time taken by insertion sort to sort data by population is less than that taken to sort data by city. Similar is the case for quick sort and merge sort.