Insertion Sort:

Insertion Sort					
Sr. No	**Run**	**File**	**Number of Record**	**Population Time**	**City(Time)**
				(sec:nanosecond)	**(sec:nanosecond)**
1	1	CENSUS2010POP-Alabama-Alabama.csv	1,102	0:7766943	0:27225278
	2			0:7934888	0:27648434
	3			0:7790758	0:27012503
	4			0:7866855	0:27192022
	5			0:7767451	0:27150558
2	1	CENSUS2010POP-Alabama-California.csv	3,920	0:77527369	0:369222141
	2			0:78168748	0:365864604
	3			0:77546485	0:362424715
	4			0:78353092	0:370565342
	5			0:78068608	0:363362056
3	1	CENSUS2010POP-Alabama-Idaho.csv	7,932	0:465865457	0:878218984
	2			0:460423153	0:930300685
	3			0:466274538	0:753019831
	4			0:468390482	0:930387992
	5			0:465283341	0:821446493
4	1	CENSUS2010POP-Alabama-Iowa.csv	21,235	2:344688079	5:323472794
	2			2:434271141	5:751041395
	3			2:152675311	5:714527051
	4			2:312099182	5:481051486
	5			2:185399812	5:557044500
5	1	CENSUS2010POP-Alabama-Missouri.csv	41,712	7:336458407	20:23207736
	2			8:281980062	20:57568978
	3			8:491600091	20:70696262
	4			8:582302111	20:20397861
	5			7:215987994	19:87481938

6	1	CENSUS2010POP.csv	81,746	26:124665176	78:168884660
	2			24:685191691	80:569225700
	3			25:682211473	80:725560541
	4			24:653082528	79:469326204
	5			26:788088370	82:593348026

- Smallest Value of CPU Time in sec:nanoseconds

Merge Sort:

Merge Sort					
Sr. No	**Run**	**File**	**Number of Record**	**Population**	**City**
				(sec:nanosecond)	**(sec:nanosecond)**
1	1	CENSUS2010POP-Alabama-Alabama.csv	1,102	0:8920681	0:9351611
	2			0:8886461	0:9388793
	3			0:8846334	0:9425770
	4			0:8898462	0:9397858
	5			0:8871832	0:9379447
2	1	CENSUS2010POP-Alabama-California.csv	3,920	0:39681342	0:42195637
	2			0:39656138	0:40076711
	3			0:39427632	0:42028067
	4			0:39478372	0:42259559
	5			0:39661967	0:41228331
3	1	CENSUS2010POP-Alabama-Idaho.csv	7,932	0:40286132	0:42785071
	2			0:39990496	0:42611221
	3			0:40124751	0:42028804
	4			0:40028804	0:42947433
	5			0:40127611	0:42686335

4	1	CENSUS2010POP-Alabama-Iowa.csv	21,235	0:158552389	0:168826275
	2			0:159138497	0:169124787
	3			0:159248612	0:169961452
	4			0:159663286	0:170350242
	5			0:158862339	0:169699841
5	1	CENSUS2010POP-Alabama-Missouri.csv	41,712	0:461923892	0:484688504
	2			0:462809272	0:489443061
	3			0:462429611	0:485718499
	4			0:463528252	0:486721343
	5			0:461636375	0:487216917
6	1	CENSUS2010POP.csv	81,746	1:517156684	1:545720104
	2			1:716002858	1:554368068
	3			1:495373215	1:565073078
	4			1:729771263	1:545313572
	5			1:494449604	1:550647201

- Smallest Value of CPU Time in sec:nanoseconds

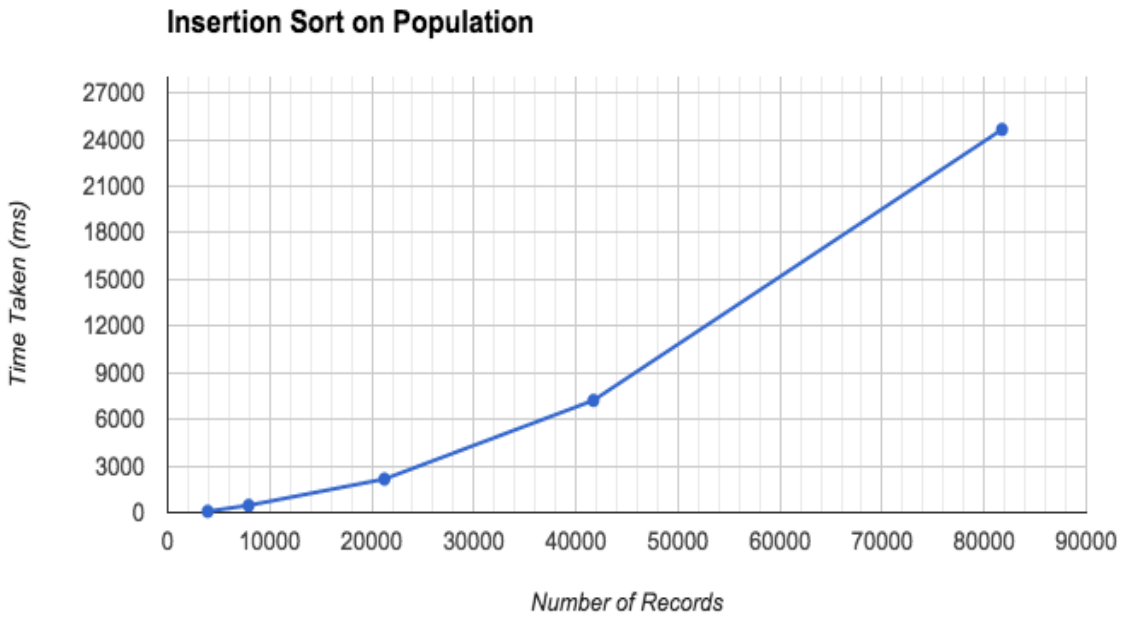
Quick Sort:

Quick Sort					
Sr. No	**Run**	**File**	**Number of Record**	**Population**	**City**
				(sec:nanosecond)	**(sec:nanosecond)**
1	1	CENSUS2010POP-Alabama-Alabama.csv	1,102	0:843411	0:1596968
	2			0:806617	0:1601192
	3			0:861684	0:1414341
	4			0:797071	0:1596741
	5			0:741103	0:1557897

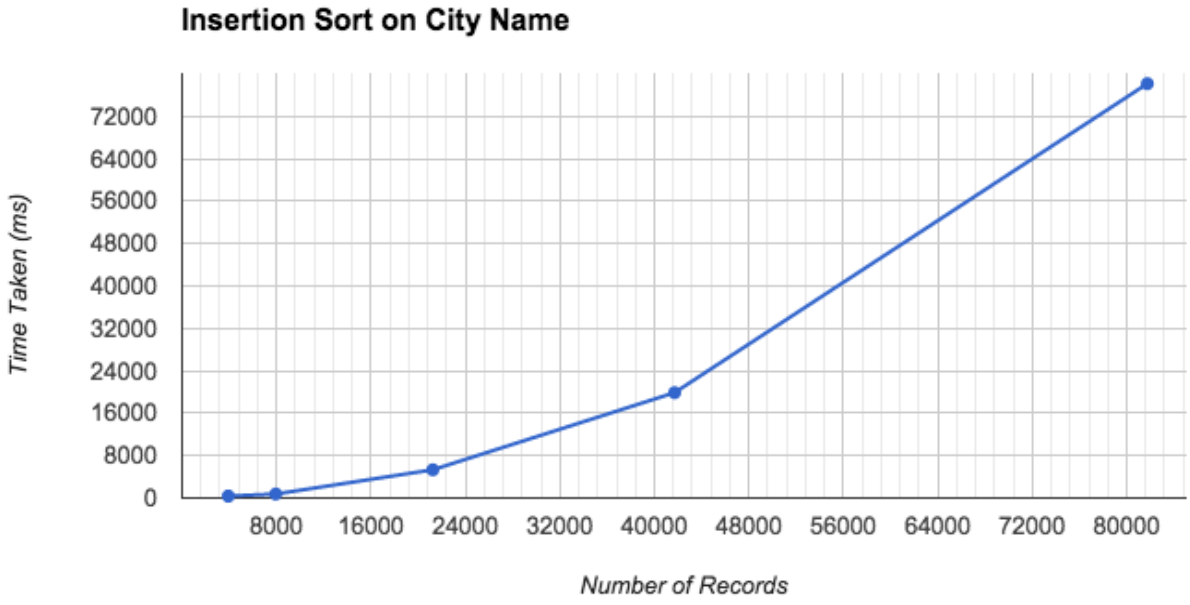
2	1	CENSUS2010POP-Alabama-California.csv	3,920	0:3320111	0:7059600
	2			0:1511478	0:2840367
	3			0:3545363	0:6706944
	4			0:3301270	0:7085303
	5			0:1641134	0:3129288
3	1	CENSUS2010POP-Alabama-Idaho.csv	7,932	0:2993268	0:6741367
	2			0:3124573	0:6600266
	3			0:2948965	0:6617111
	4			0:3090145	0:6980151
	5			0:3065084	0:6624736
4	1	CENSUS2010POP-Alabama-Iowa.csv	21,235	0:11041574	0:22377122
	2			0:11105157	0:22945521
	3			0:10862993	0:23377506
	4			0:10985307	0:22225368
	5			0:10577104	0:23432666
5	1	CENSUS2010POP-Alabama-Missouri.csv	41,712	0:26353062	0:47173516
	2			0:26177009	0:48495061
	3			0:27030829	0:48137193
	4			0:26952984	0:49883631
	5			0:26738159	0:47413792
6	1	CENSUS2010POP.csv	81,746	0:65561794	0:118314248
	2			0:68280941	0:113466821
	3			0:68125535	0:127098771
	4			0:69069511	0:117806948
	5			0:64279311	0:147493654

GRAPHS:

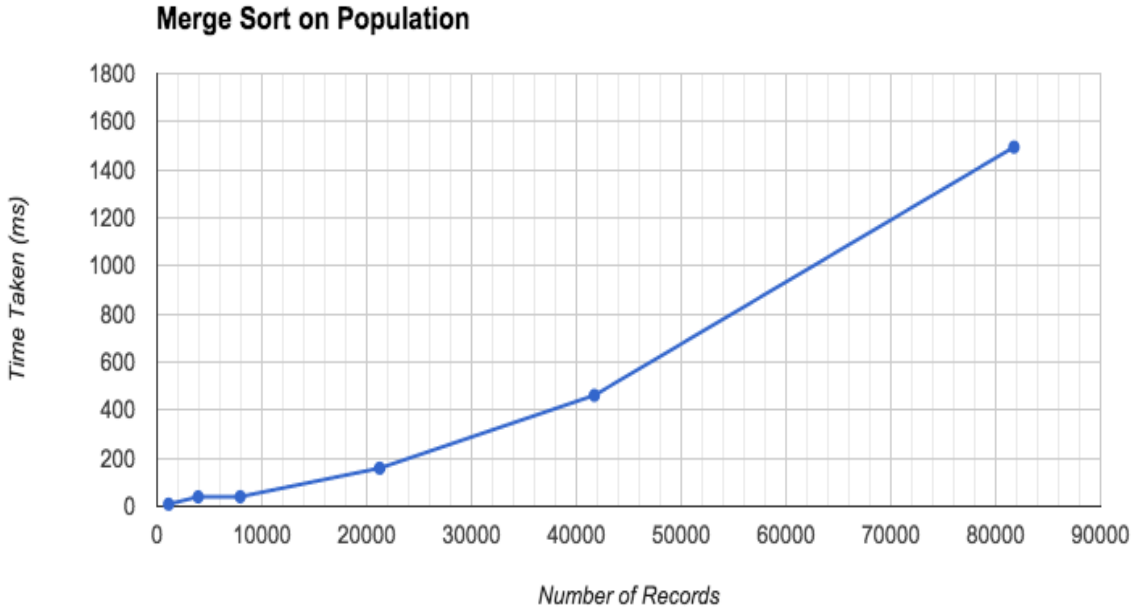
Insertion Sort on Population	
Number of Records	**Time Taken(ms)**
1102	7.766943
3920	77.527369
7932	460.423153
21235	2152.675311
41712	7215.987994
81746	24653.08253



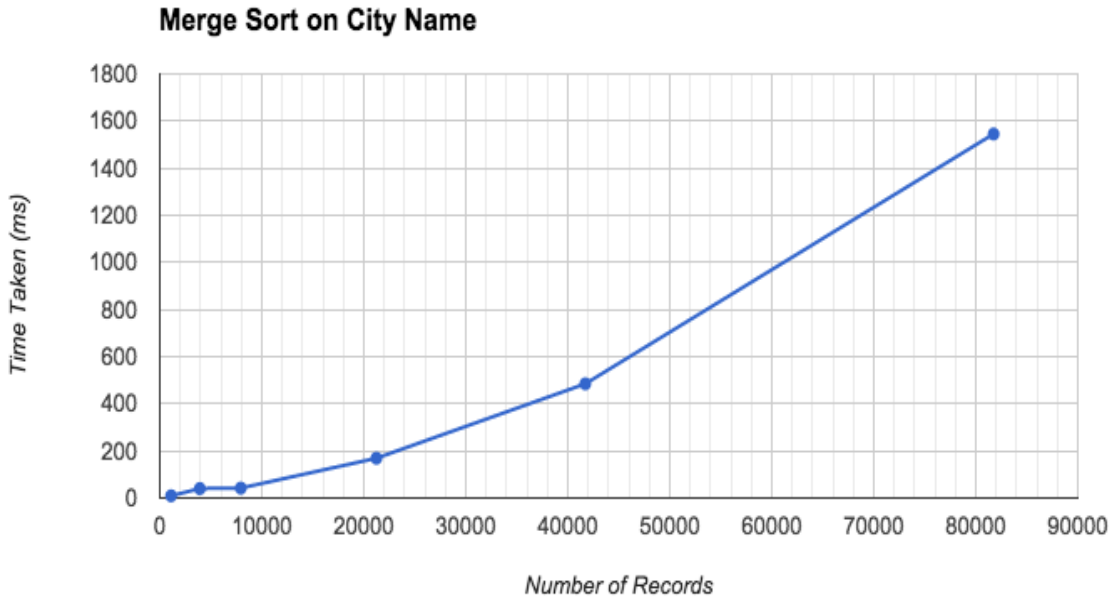
Insertion sort on City Name	
Number of Records	**Time Taken(ms)**
1102	27.012503
3920	362.424715
7932	753.019831
21235	5323.472794
41712	19874.81939
81746	78168.88466



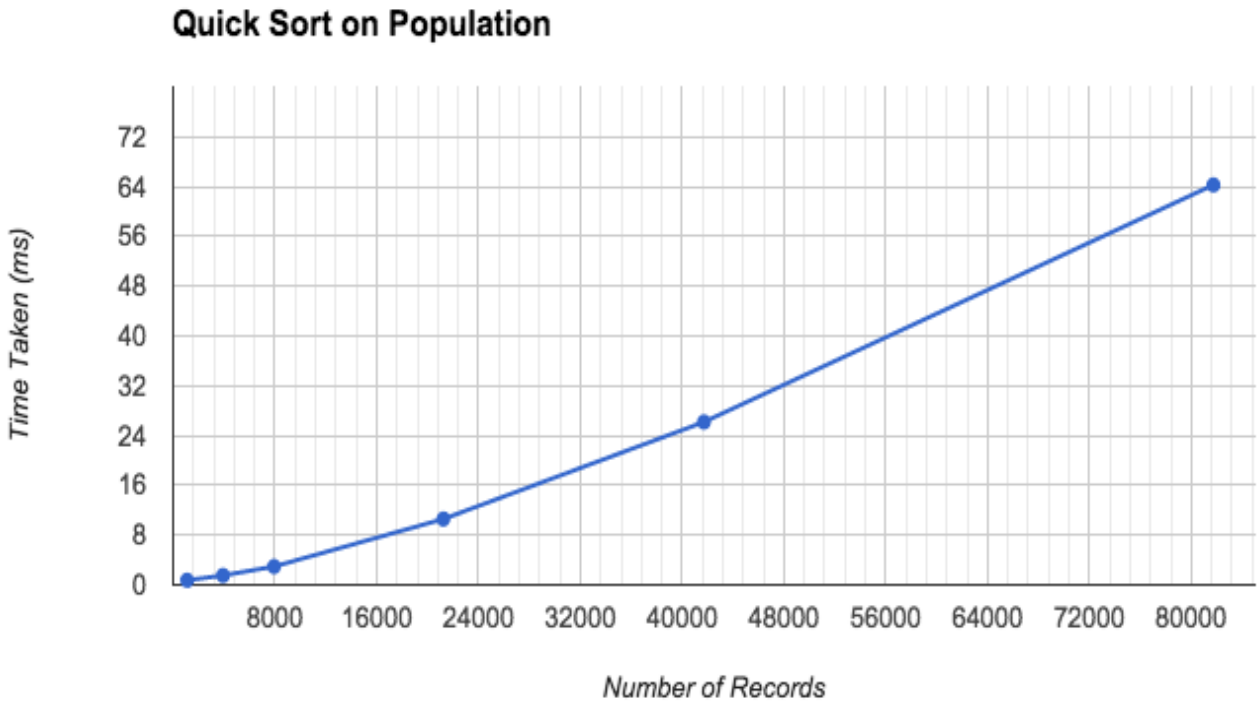
Merge Sort on Population	
Number of Records	**Time Taken(ms)**
1102	8.846334
3920	39.427632
7932	39.990496
21235	158.552389
41712	461.636375
81746	1494.449604



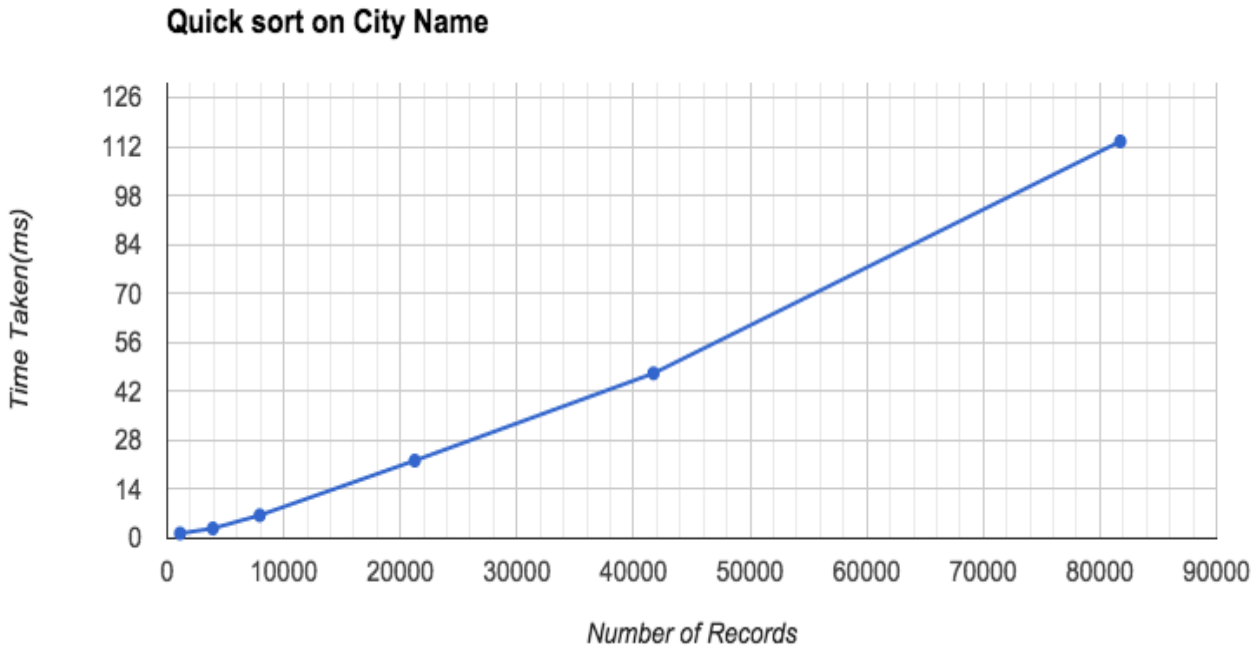
Merge Sort on City Name	
Number of Records	**Time Taken(ms)**
1102	9.351611
3920	40.076711
7932	42.028804
21235	168.826275
41712	484.688504
81746	1545.313572



Quick Sort on Population	
Number of Records	**Time Taken(ms)**
1102	0.741103
3920	1.511478
7932	2.948965
21235	10.577104
41712	26.177009
81746	64.279311



Quick Sort on City Name	
Number of Records	**Time Taken(ms)**
1102	1.414341
3920	2.840367
7932	6.600266
21235	22.225368
41712	47.173516
81746	113.466821



Conclusion:

By looking at the tables and graphs which are being obtained we can draw following conclusions:

Insertion Sort:

1. Insertion sort takes a lot of time for sorting data with large number of records(input size). We see that the time taken increases as we increase the number of inputs(records).
2. The output of insertion sort is stable. The relative order of output according to the input does not change for records with same values (Observed from output files).
3. The point line graph is between linear and quadratic, but tends more towards quadratic line.
4. The time taken for insertion sort on name(String) is approximately three times as that of population(integer). This shows that the insertion sort performance is largely dependent on type of input data.
5. Insertion sort performance is largely dependent on input size in contrast to Merge and Quick sort. The performance is worst compared to merge and quick sort as we increase the size of inputs. Hence, Insertion is least suitable for inputs with large input size.

Merge Sort:

1. Merge sort performance is much better than that of insertion sort for the large input size but not as good as quick sort.
2. The output of merge sort is stable. The relative order of output according to the input does not change for records with same values. (Observed from output files).
3. The point line graph is approximately linearithmic.
4. The point graph line is between linear and quadratic, more towards quadratic.
5. The time taken for merge sort on city name(String) is slightly greater than that of merge sort on population(Integer). We can say that merge sort is not completely dependent on the type of input.

Quick Sort:

1. Quick sort performance is the best amongst all the three types of sorts.
2. The output in the file is not stable. The relative order of output is not according to the input for records with same values.
3. The point line graph is approximately linearithmic.
4. Quick sort on population(Integer) is faster (almost double) than the quick sort on name(String).
5. In the table for quick sort we can see that the performance is not approximately same. Quick sort performance depends on the pivot element which is being selected and hence the performance varies according to it.
6. Quick sort is slightly faster than Merge Sort and largely faster than Insertion sort. Hence Quick Sort is best suitable for the large size of data and independent of type of input data.