Report- Programmong Assignment : 02

Table containing time required to sort given data for Insertion sort, Merge Sort and Quick Sort. Each file is given as a input for program and executed five times. The following table contents time required to sort and then I have highlighted the minimum time required to sort from it.

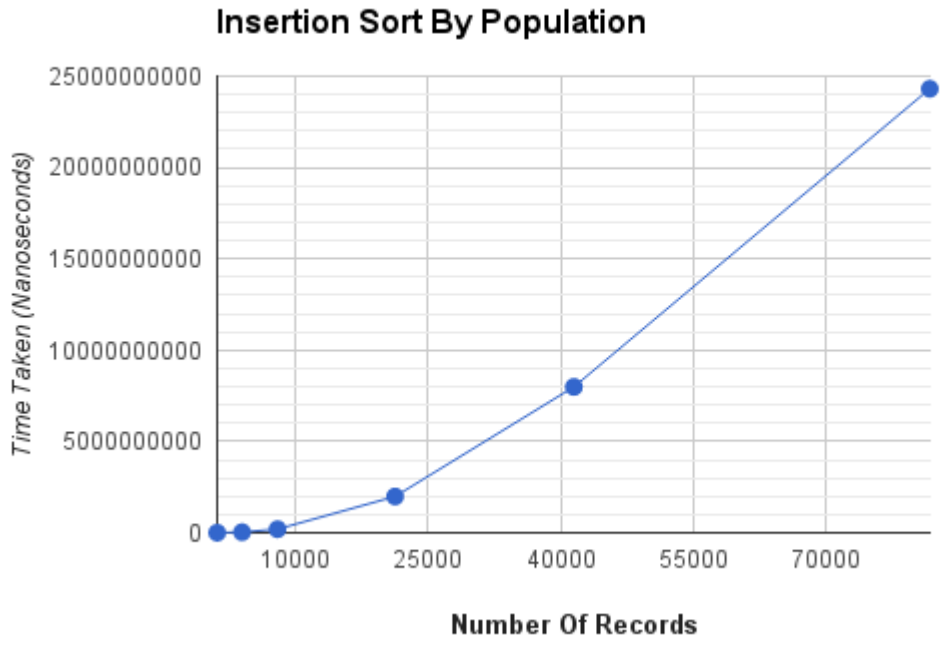
	Insertion Sort		**Merge Sort**		**Quick sort**	
	Time to sort by Population	Time to sort By Name	Time to sort by Population	Time to sort By Name	Time to sort by Population	Time to sort By Name
Run1_Alabama	0:3323373	0:12136238	0:1870391	0:2086354	0:396118	0:681168
Run2_Alabama	0:3315385	0:11880595	0:1865714	0:2076514	0:399716	0:722786
Run3_Alabama	0:3272503	0:11520970	0:1872351	0:2058137	0:391980	0:703914
Run4_Alabama	0:3323439	0:12118816	0:1868054	0:2055355	0:413259	0:679026
Run5_alabama	0:3277908	0:11942095	0:1892498	0:2074859	0:394804	0:687360
Min	0:3272503	0:11520970	0:1865714	0:2055355	0:394804	0:679026
Run1_california	0:33302124	0:163915128	0:6943475	0:7887852	0:1635217	0:3450355
Run2_california	0:33144641	0:162837216	0:6836973	0:8040320	0:1526684	0:3189214
Run 3_california	0:33151798	0:160541651	0:6841815	0:7895317	0:1551950	0:3174986
Run 4_california	0:33152454	0:163325139	0:6909917	0:8048828	0:1574317	0:3587430
Run 5_california	0:33658763	0:165398150	0:6910370	0:8018374	0:1574407	0:3325569
Min	0:33302124	0:160541651	0:6836973	0:7887852	0:1526684	0:3174986
Run 1_Idaho	0:197997190	0:760091006	0:14538113	0:17753316	0:3772986	0:7725018
Run 2_Idaho	0:198378392	0:757055507	0:14987946	0:18567593	0:3573002	0:8243822
Run 3_Idaho	0:198175027	0:758590353	0:15119185	0:18353802	0:3621604	0:8269499
Run 4_Idaho	0:198839654	0:753835968	0:14732507	0:18228594	0:3643575	0:9199106
Run 5_Idaho	0:198242118	0:753300033	0:14610121	0:17463027	0:3609041	0:8300488
Min	0:197997190	0:753835968	0:14538113	0:17463027	0:3573002	0:7725018

Run 1_Iowa	2:6419433	6:579176140	0:45182108	0:60794955	0:13011560	0:29838672
Run 2_Iowa	1:986562341	6:173600414	0:44256677	0:58904174	0:13576384	0:33246395
Run 3_Iowa	2:37313151	5:996691774	0:43932436	0:57754085	0:12882897	0:30793413
Run 4_Iowa	2:7059371	5:904260144	0:43140174	0:55438144	0:12562437	0:26472474
Run 5_Iowa	2:7658179	5:925685320	0:41826504	0:52009661	0:12740789	0:23380517
Minimum	1:986562341	5:904260144	0:41826504	0:52009661	0:12562437	0: 23380517
Run 1_Missouri	8:271722625	31:200059178	0:89883750	0:125222280	0:33140161	0:78486188
Run 2_Missouri	8:453695852	26:823877510	0:90181762	0:123755718	0:32947203	0:75749384
Run 3_Missouri	8:95400070	24:304713277	0:87473341	0:115601270	0:31557386	0:62913070
Run 4_Missouri	7:987348113	20:719538418	0:87654447	0:115426851	0:31850654	0:62660814
Run 5_Missouri	8:213950019	22:655399286	0:84879960	0:107717429	0:30244856	0:54123921
Min	7:987348113	20:719538418	0:84879960	0:107717429	0:30244856	0:54123921
Run 1_Largefile	24:309588407	141:967041133	0:181852266	0:247541845	0:78918765	0:156010123
Run 2_Largefile	25:11497272	104:152964700	0:181308645	0:243910660	0:78272329	0:142607501
Run 3_Largefile	31:389459265	127:139808633	0:178868881	0:245342113	0:77206852	0:161576876
Run 4_Largefile	31:863134889	138:818271781	0:178635881	0:254605440	0:80282860	0:176463973
Run 5_Largefile	28:680732219	89:803667602	0:175273914	0:229231819	0:74568439	0:140921504
Minimun	24:309588407	104:152964700	0:175273914	0:229231819	0:74568439	0:140921504

Analysis of Insertion Sort:

1. Insertion Sort By Population:

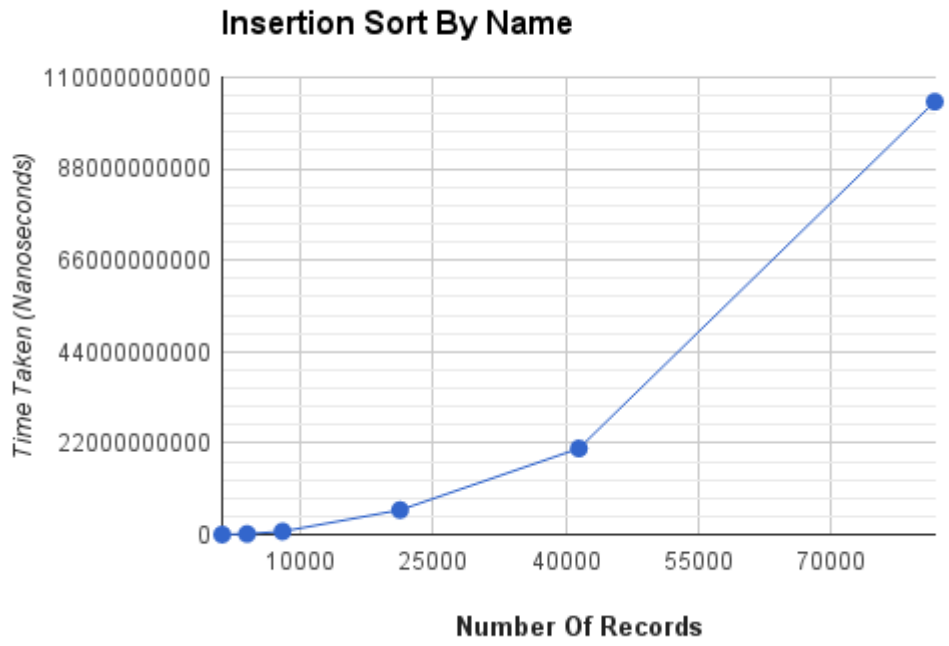
Number of records	Time Taken (min)
1102	3272503
3920	33302124
7932	197997190
21236	1986562340
41472	7987348113
81746	24309588407



In insertion Sort by population, we have sorted record with respect to value of population and input is an integer values. Time required to sort input of different size is varies as number of inputs increases. It requires time less than Insertion sort by name and more time than merge sort and quick sort.

2. Insertion Sort By Name:

Number of records	Time Taken (min)
1102	11520970
3920	160541651
7932	753835968
21236	5904260144
41472	20719538418
81746	104152964699



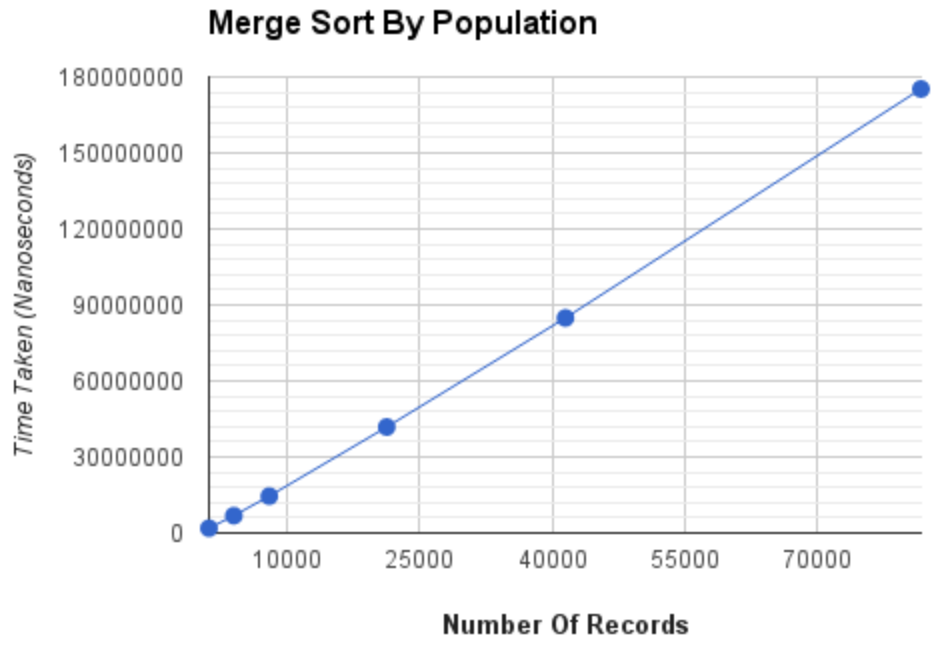
In insertion Sort by Name, we have sorted record with respect to value of name of city and input is an character. Time required to sort input increases as number of inputs increases. It requires time more than insertion sort by population and more time than merge and quick sort but more than Insertion Sort By Polulation.

Eg. Insertion_By Population =24309588407 >>>Insertion_ By name=104152964699

Analysis of Merge Sort

1. Merge Sort By Population:

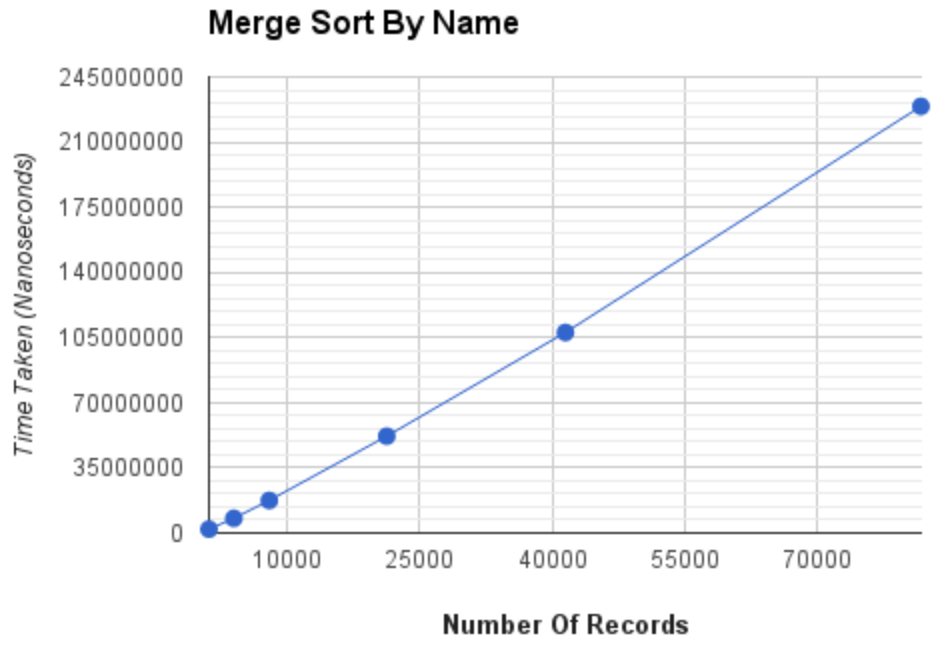
Number of records	Time Taken(min)
1102	1865714
3920	6836973
7932	14538113
21236	41826504
41472	84879960
81746	175273914



In Merge Sort by population, we have sorted record with respect to value of population of city and input is an integer. It requires time less than Insertion and quick sort. We use Divide and conquer algorithm to reduce running time of merge sort.

2. Merge Sort By Name:

Number of records	Time Taken(min)
1102	2055355
3920	7887852
7932	17463027
21236	52009661
41472	107717429
81746	229231819



In Merge Sort by Name, we have sorted record with respect to Ascii value of name of city and input is an integer. Times required to sort input of different is varies as number of inputs increases. It requires time less than Insertion sort and quick sort .

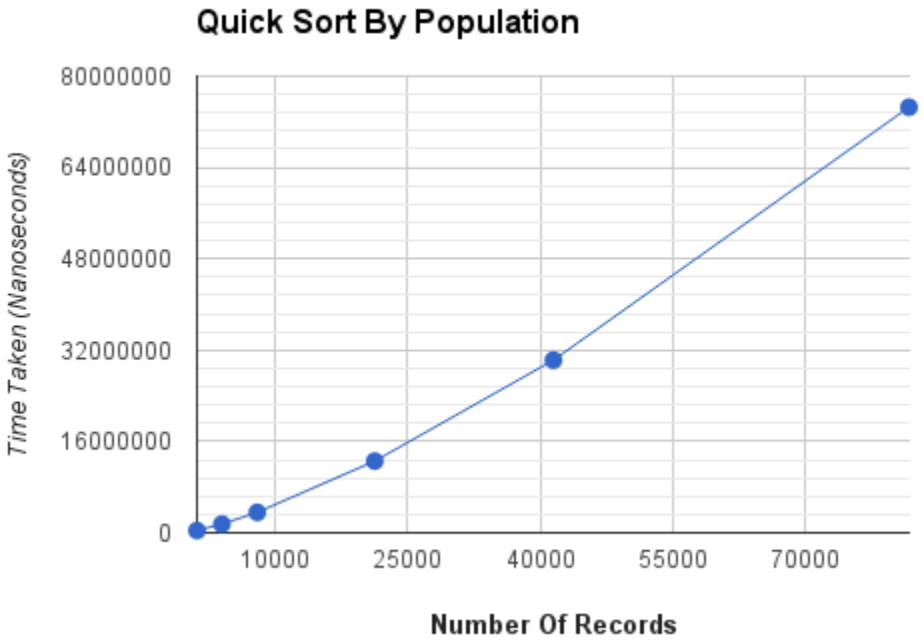
Merge sort is the efficient algorithm form (insertion,merge and quick)

Eg. Merge_By Population =175273914 >>> merge_By name=229231819

Analysis of Quick Sort

1. Quick Sort By Population:

Number of records	Time Taken(min)
1102	394804
3920	1526684
7932	3573002
21236	12562437
41472	30244856
81746	74568439

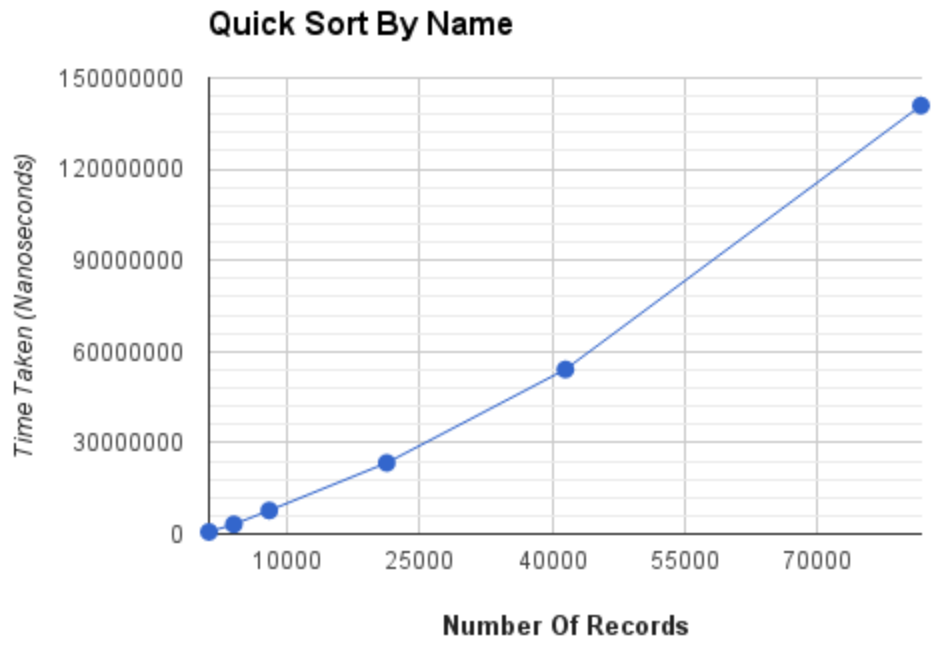


Quick sort is less efficient algorithm than Merge Sort but more efficient than insertion sort. Time required for execution is increase as number if input increases as represented in above graph.

\

2. Quick Sort By Name:

Number of records	Time Taken(min)
1102	679026
3920	3174986
7932	7725018
21236	23380517
41472	54123921
81746	140921504



Quick sort by name required more time than quick sort by population.

Eg. Quick_By Population =74568439 > Quick_By name=140921504

Conclusion: In these three algorithms merge sort is the more efficient algorithm Merge Sort takes less time of execution. Merge sort includes divide and conquer algorithm to reduce time of execution.