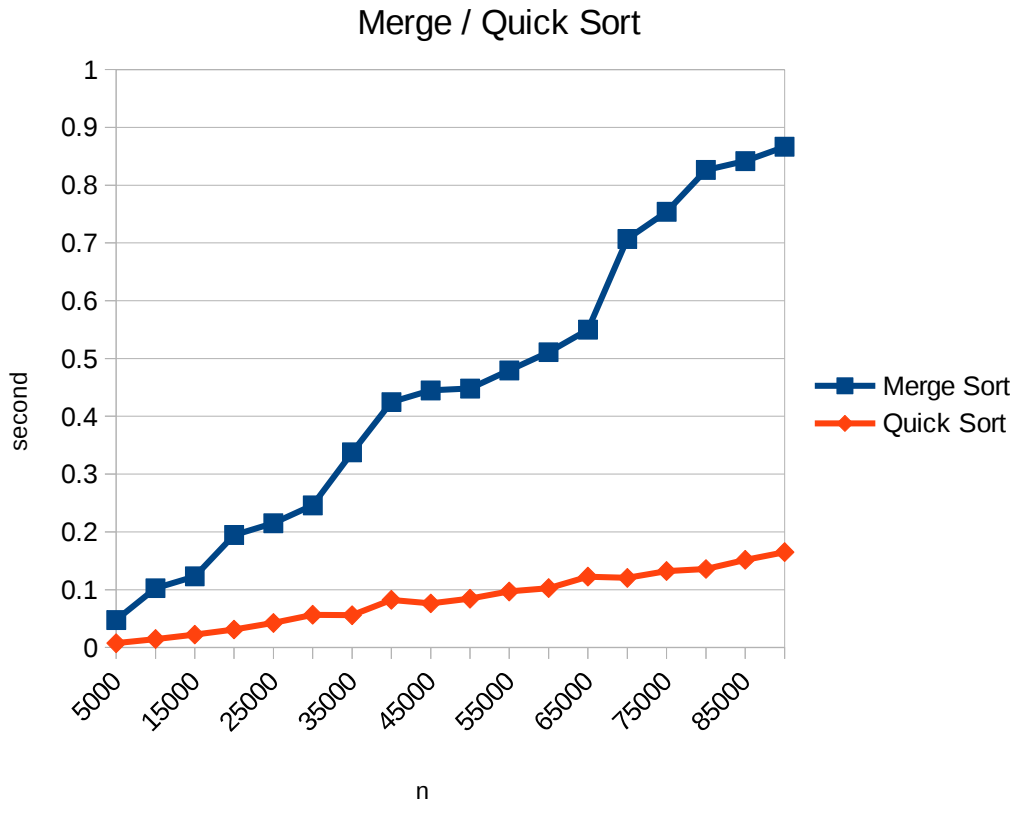
Name: Son Nguyen
Email: snguyen313@csu.fullerton.edu
Project #2: Merge sort and Quick sort

1 Plot and draw chart

	Elapsed Time (sec)		Compare Quick/ Merge Sort
n	Merge Sort	Quick Sort	
5000	0.0475248	0.00741819	15.61%
10000	0.102637	0.014629	14.25%
15000	0.123052	0.0223291	18.15%
20000	0.194585	0.031026	15.94%
25000	0.215059	0.0426009	19.81%
30000	0.245931	0.0565052	22.98%
35000	0.337586	0.0559638	16.58%
40000	0.424739	0.0823688	19.39%
45000	0.444991	0.0763649	17.16%
50000	0.448156	0.0845029	18.86%
55000	0.479541	0.097043	20.24%
60000	0.511006	0.102676	20.09%
65000	0.54998	0.122495	22.27%
70000	0.706988	0.120523	17.05%
75000	0.753815	0.132308	17.55%
80000	0.826313	0.135855	16.44%
85000	0.84176	0.151667	18.02%
90000	0.866507	0.164987	19.04%



- Looking at the above chart, we can see elapsed time of merge sort is represented by an linear line which efficiency is almost O(nlogn) as stated by the lecture
- For Quick sort, even though the lecture says that its efficiency is O(nlogn) for the best case,

and O(n2) for the worst case, the chart and the data table shows that its elapsed time is just around 20% of merge sort

2 Calculate average and Standard Deviation

	Merge Sort		Quick Sort		
n	Elapsed Time (sec)	Convert to Average of elapsed time for n = 5000	Elapsed Time (sec)	Convert to Average of elapsed time for n = 5000	Compare Quick/ Merge Sort
5000	0.0480606	0.0480606	0.00710619	0.00710619	14.79%
5000	0.0407481	0.0407481	0.00662842	0.00662842	16.27%
10000	0.0936631	0.04683155	0.0144574	0.0072287	15.44%
10000	0.0907357	0.04536785	0.0143262	0.0071631	15.79%
15000	0.123609	0.041203	0.0227012	0.0075670667	18.37%
15000	0.120517	0.0401723333	0.0216277	0.0072092333	17.95%
20000	0.208888	0.052222	0.0317301	0.007932525	15.19%
20000	0.183946	0.0459865	0.0305961	0.007649025	16.63%
25000	0.218798	0.0437596	0.0405833	0.00811666	18.55%
25000	0.212371	0.0424742	0.0414297	0.00828594	19.51%
30000	0.264785	0.0441308333	0.0486829	0.0081138167	18.39%
30000	0.24913	0.0415216667	0.0508074	0.0084679	20.39%
35000	0.364518	0.052074	0.0591044	0.0084434857	16.21%
35000	0.335008	0.0478582857	0.0571242	0.0081606	17.05%
40000	0.392464	0.049058	0.0650352	0.0081294	16.57%
40000	0.387704	0.048463	0.0661597	0.0082699625	17.06%
45000	0.421082	0.0467868889	0.0753122	0.0083680222	17.89%
45000	0.418195	0.0464661111	0.0835973	0.0092885889	19.99%
50000	0.471682	0.0471682	0.0852247	0.00852247	18.07%
50000	0.473184	0.0473184	0.0841977	0.00841977	17.79%
55000	0.49663	0.0451481818	0.0993635	0.0090330455	20.01%
55000	0.490415	0.0445831818	0.0970004	0.0088182182	19.78%
60000	0.527863	0.0439885833	0.105469	0.0087890833	19.98%
60000	0.524755	0.0437295833	0.0988057	0.0082338083	18.83%
65000	0.558956	0.0429966154	0.113997	0.008769	20.39%
65000	0.542886	0.0417604615	0.114004	0.0087695385	21.00%
70000	0.738259	0.0527327857	0.122033	0.0087166429	16.53%
70000	0.718521	0.0513229286	0.118755	0.0084825	16.53%
75000	0.799301	0.0532867333	0.132015	0.008801	16.52%
75000	0.785767	0.0523844667	0.133029	0.0088686	16.93%
80000	0.809604	0.05060025	0.157186	0.009824125	19.42%
80000	0.829494	0.051843375	0.152998	0.009562375	18.44%
85000	0.866979	0.0509987647	0.15611	0.0091829412	18.01%
85000	0.87087	0.0512276471	0.147486	0.0086756471	16.94%
90000	0.899584	0.0499768889	0.171124	0.0095068889	19.02%
90000	0.885482	0.0491934444	0.174235	0.0096797222	19.68%
Average		**0.0470401392**		**0.008410667**	**17.88%**
Min		**0.0401723333**		**0.00662842**	**16.50%**
Max		**0.0532867333**		**0.009824125**	**18.44%**
Stdev		**0.003893027**		**0.0007594265**	**19.51%**

- As shown in the above table, Average of elapsed time of merge sort for n = 5000 is 0.047s and look at the detail in each run (for n = 5000 .. 90000,) the elapsed times are converged around 0.4 ~ 0.5 which proves what lecture stated: Efficiency is always O(nlogn) for both best and worst case
- Standard deviation of the merge sort's elapsed time is relative small (~0.004) which means the time for every run is always converged near the average value (the difference between

min and max is very small (~0.013)
- For quick sort, even though the average of elapsed time is smaller than merge sort, we can see the average time increases faster than merge sort when n increased