ASCENDING	Insertion	Bubble	Selection		Quick	Merge
N=50000				N=500000		
Trial 1	0.000893	0.000216	2.81378	Trial 1	0.032088	0.081636
Trial 2	0.000808	0.000386	2.55934	Trial 2	0.044354	0.086208
Trial 3	0.000807	0.000562	2.63204	Trial 3	0.045204	0.093128
Average	0.000836	0.000388	2.668387	Average	0.040549	0.086991
N=100000				N=1000000		
Trial 1	0.001696	0.000701	10.2581	Trial 1	0.098173	0.124276
Trial 2	0.00161	0.001005	10.1433	Trial 2	0.071426	0.155342
Trial 3	0.001619	0.00112	10.2172	Trial 3	0.094972	0.134638
Average	0.001642	0.000942	10.2062	Average	0.08819	0.138085
N=200000				N=2000000		
Trial 1	0.001964	0.00194	40.9844	Trial 1	0.134492	0.205823
Trial 2	0.002084	0.002289	40.9052	Trial 2	0.151569	0.205635
Trial 3	0.002796	0.001996	40.6289	Trial 3	0.141811	0.206165
Average	0.002281	0.002075	40.8395	Average	0.142624	0.205874
N=300000				N=4000000		
Trial 1	0.003521	0.002193	91.3997	Trial 1	0.177861	0.468442
Trial 2	0.00432	0.003277	91.7247	Trial 2	0.221913	0.443342
Trial 3	0.00286	0.001765	91.5082	Trial 3	0.209582	0.428878
Average	0.003567	0.002412	91.5442	Average	0.203119	0.446887
N=400000				N=8000000		
Trial 1	0.005322	0.003928	163.275	Trial 1	0.375416	0.893485
Trial 2	0.008047	0.003019	162.81	Trial 2	0.387257	0.891984
Trial 3	0.003592	0.002568	162.738	Trial 3	0.425358	0.894704
Average	0.005654	0.003172	162.941	Average	0.39601	0.893391

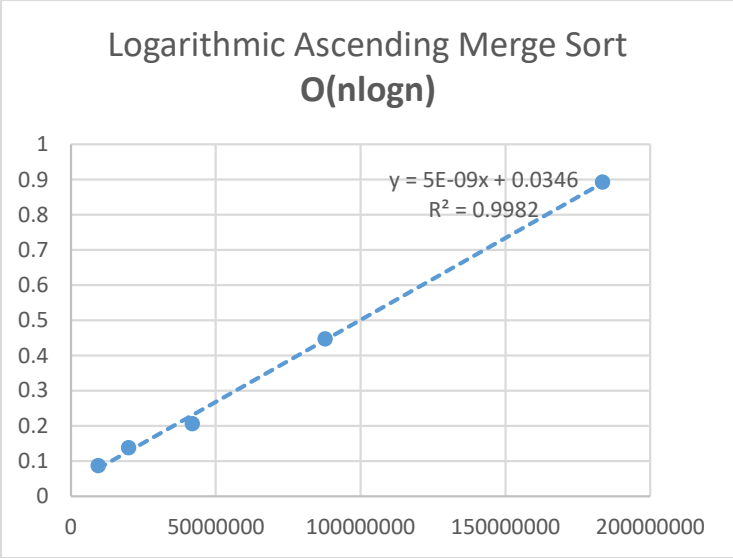
DESCENDING	Insertion	Bubble	Selection		Quick	Merge
N=50000				N=500000		
Trial 1	3.63327	12.5636	2.73254	Trial 1	0.102932	0.099919
Trial 2	3.59967	12.6725	2.69035	Trial 2	0.043863	0.080435
Trial 3	3.57632	12.5421	2.71377	Trial 3	0.072941	0.086499
Average	3.603087	12.59273	2.71222	Average	0.073245	0.088951
N=75000				N=1000000		
Trial 1	8.12066	28.2384	6.05225	Trial 1	0.14953	0.162726
Trial 2	8.05223	28.1151	6.01711	Trial 2	0.129775	0.174267
Trial 3	8.20243	28.1068	6.12157	Trial 3	0.109296	0.09885
Average	8.125107	28.15343	6.063643	Average	0.129534	0.145281
N=100000				N=2000000		
Trial 1	14.4003	50.0188	10.7616	Trial 1	0.211382	0.242709
Trial 2	14.3893	55.7995	10.7599	Trial 2	0.191602	0.205352
Trial 3	14.449	50.0044	10.6929	Trial 3	0.20075	0.206515
Average	14.41287	51.9409	10.73813	Average	0.201245	0.218192
N=125000				N=4000000		
Trial 1	22.3221	82.46	16.7379	Trial 1	0.363897	0.4309
Trial 2	22.3549	78.3356	16.8473	Trial 2	0.399091	0.446067
Trial 3	22.4042	78.1765	16.7597	Trial 3	0.375588	0.427509
Average	22.3604	79.65737	16.78163	Average	0.379525	0.434825
N=150000				N=8000000		
Trial 1	32.6394	112.964	24.1541	Trial 1	0.771117	0.917299
Trial 2	32.3632	113.13	24.2058	Trial 2	0.755377	0.889698
Trial 3	32.2676	113.423	24.1297	Trial 3	0.767636	0.892223

Average	32.4234	113.1723	24.1632	Average	0.76471	0.89974
RANDOM	Insertion	Bubble	Selection		Quick	Merge
N=50000				N=500000		
Trial 1	1.84076	12.0542	2.85693	Trial 1	0.153533	0.144595
Trial 2	1.84207	12.1433	2.82463	Trial 2	0.092047	0.129192
Trial 3	1.84156	11.7733	2.83329	Trial 3	0.147916	0.138377
Average	1.841463	11.99027	2.838283	Average	0.131165	0.137388
N=75000				N=1000000		
Trial 1	4.0471	26.5201	6.33825	Trial 1	0.212111	0.261591
Trial 2	4.10224	26.5403	6.33871	Trial 2	0.221816	0.203607
Trial 3	4.10646	26.5469	6.33456	Trial 3	0.169798	0.21392
Average	4.085267	26.53577	6.337173	Average	0.201242	0.226373
N=100000				N=2000000		
Trial 1	7.2261	47.2062	11.3481	Trial 1	0.348157	0.420192
Trial 2	7.29068	46.9308	11.2488	Trial 2	0.336985	0.434883
Trial 3	7.37978	47.1627	11.3049	Trial 3	0.334415	0.418753
Average	7.298853	47.0999	11.3006	Average	0.339852	0.424609
N=125000				N=4000000		
Trial 1	11.7165	74.2431	17.6128	Trial 1	0.70215	0.847799
Trial 2	11.4327	73.8016	17.628	Trial 2	0.705368	0.84962
Trial 3	11.5872	73.5611	17.7921	Trial 3	0.701429	0.846143
Average	11.5788	73.8686	17.67763	Average	0.702982	0.847854
N=150000				N=8000000		
Trial 1	16.5181	106.097	25.5351	Trial 1	1.45214	1.78434
Trial 2	16.4305	105.706	25.39	Trial 2	1.44642	1.76428
Trial 3	16.7354	106.954	25.4319	Trial 3	1.44973	1.78997
Average	16.56133	106.2523	25.45233	Average	1.44943	1.77953

Merge Sort

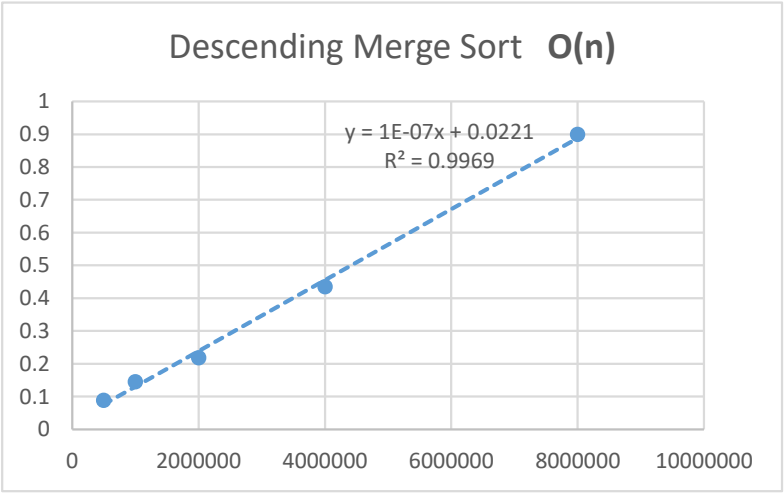
Ascending

500000	9465784.28	0.08699067
1000000	19931568.6	0.13808533
2000000	41863137.1	0.20587433
4000000	87726274.3	0.44688733
8000000	183452549	0.893391



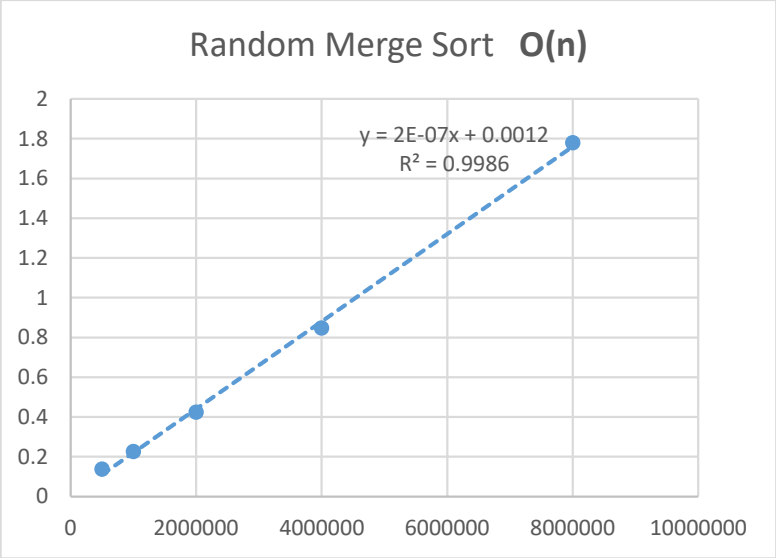
Descending

500000	0.088951
1000000	0.145281
2000000	0.218192
4000000	0.43482533
8000000	0.89974



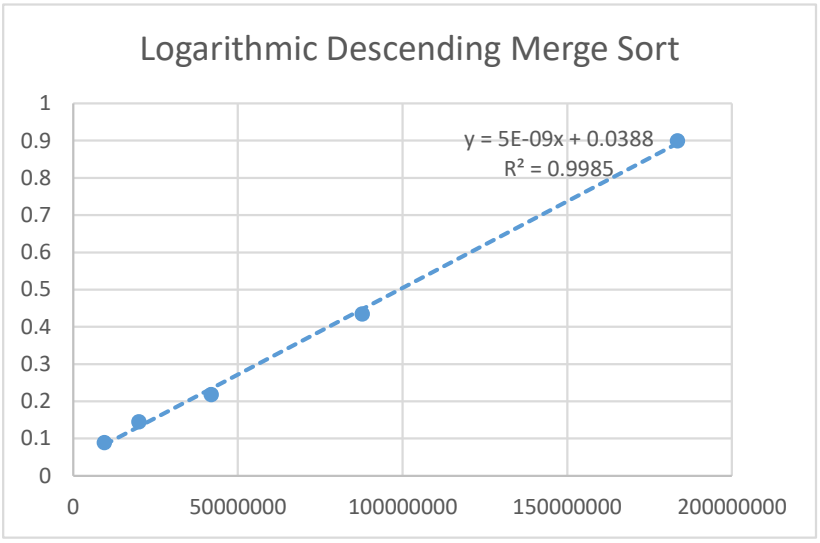
Random

500000	0.137388
1000000	0.22637267
2000000	0.42460933
4000000	0.847854
8000000	1.77953



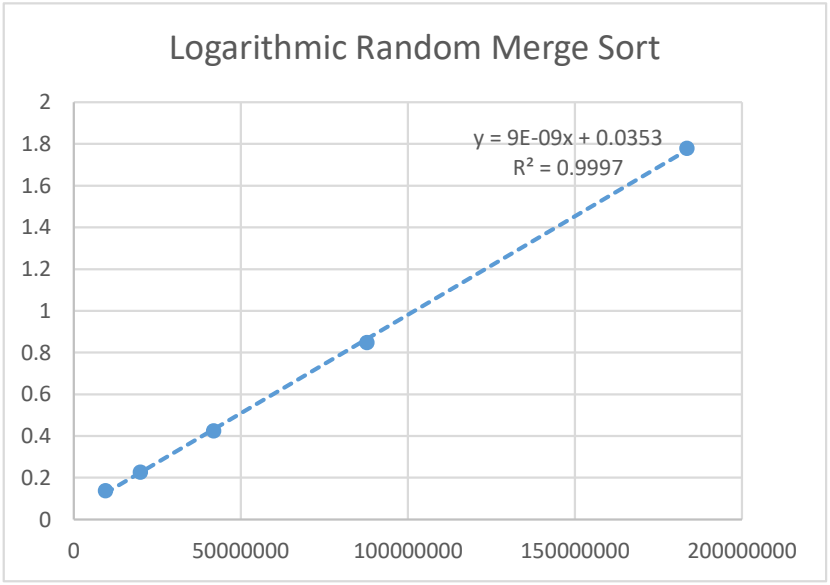
9465784.28	0.088951
19931568.6	0.145281
41863137.1	0.218192
87726274.3	0.43482533
183452549	0.89974

This column is N*Log_2(N)



9465784.28	0.137388
19931568.6	0.22637267
41863137.1	0.42460933
87726274.3	0.847854
183452549	1.77953

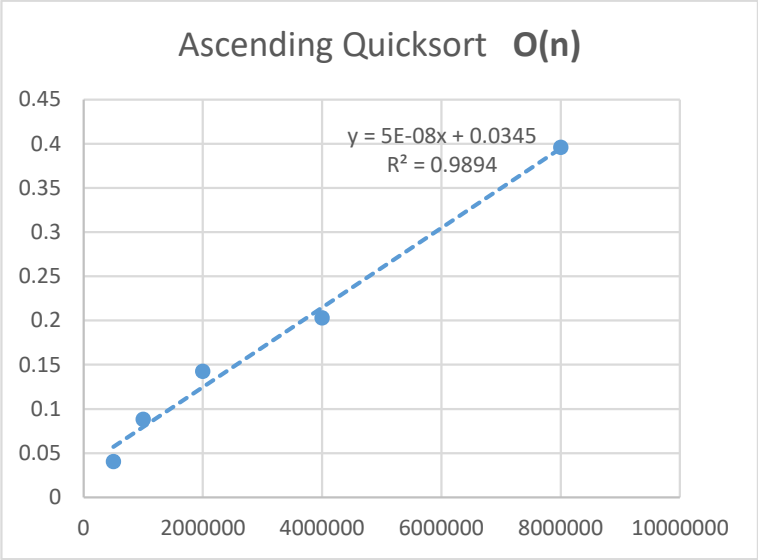
This column is N*Log_2(N)



Quick Sort

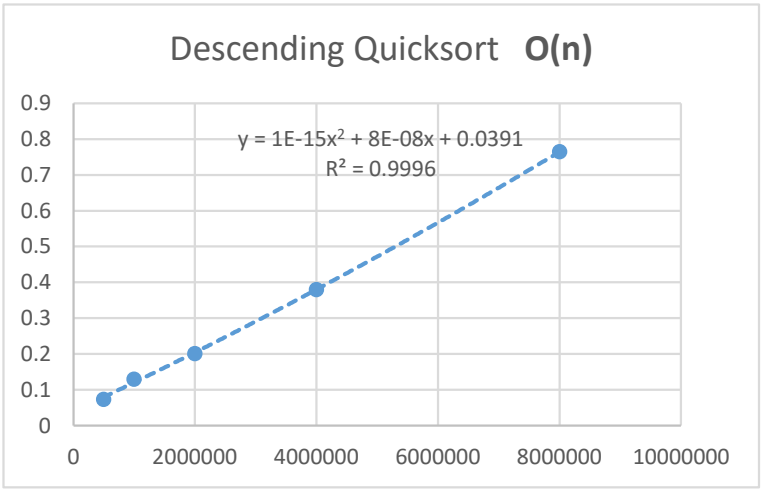
Ascending

500000	0.04054867
1000000	0.08819033
2000000	0.142624
4000000	0.20311867
8000000	0.39601033



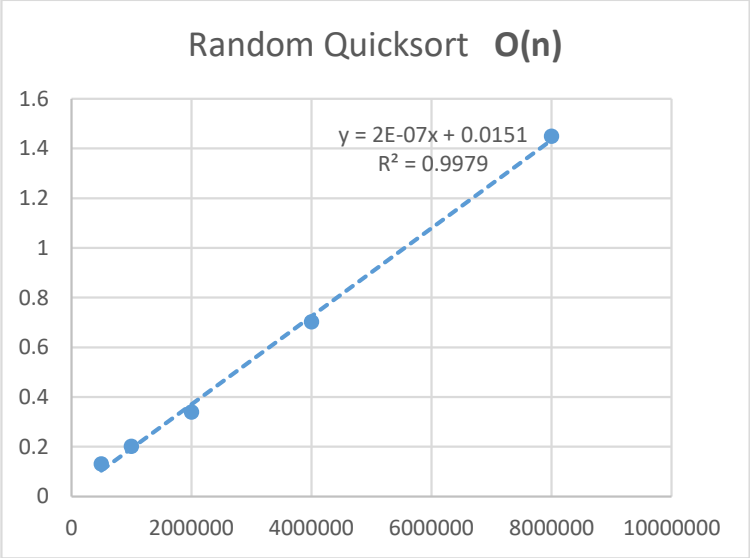
Descending

500000	0.07324533
1000000	0.12953367
2000000	0.20124467
4000000	0.37952533
8000000	0.76471



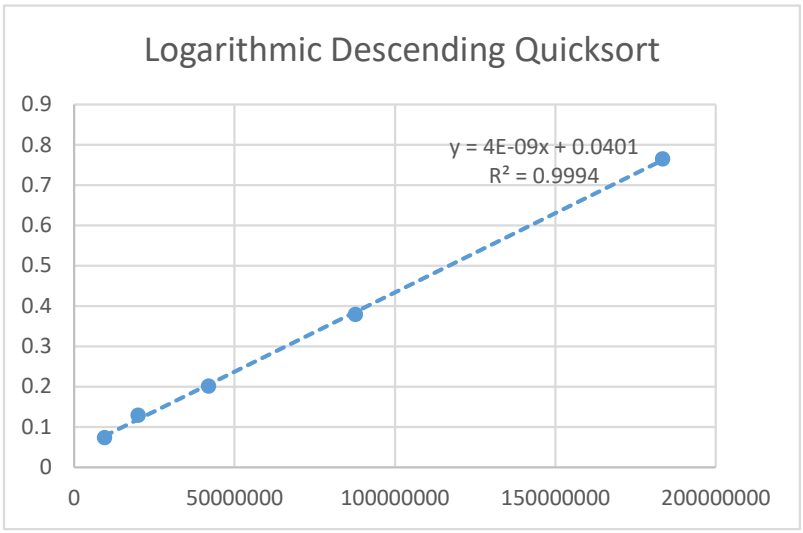
Random

500000	0.13116533
1000000	0.20124167
2000000	0.33985233
4000000	0.70298233
8000000	1.44943



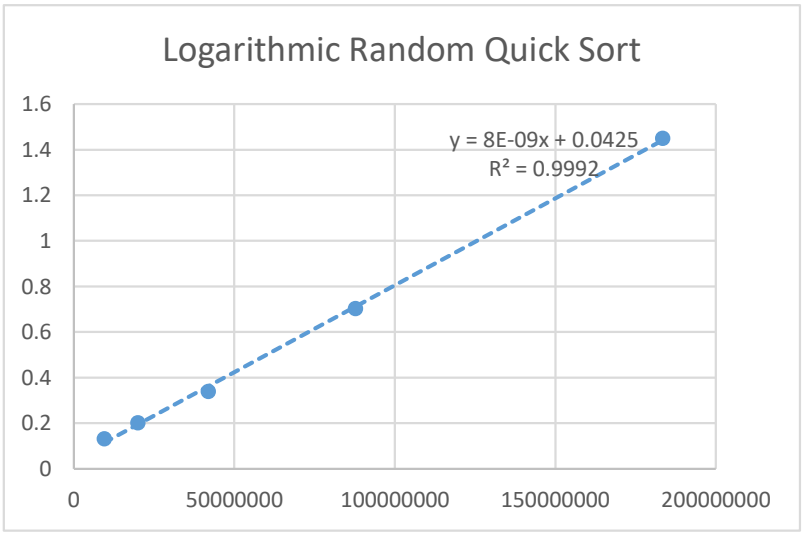
9465784.28	0.07324533
19931568.6	0.12953367
41863137.1	0.20124467
87726274.3	0.37952533
183452549	0.76471

This column is N*Log_2(N)



9465784.28	0.13116533
19931568.6	0.20124167
41863137.1	0.33985233
87726274.3	0.70298233
183452549	1.44943

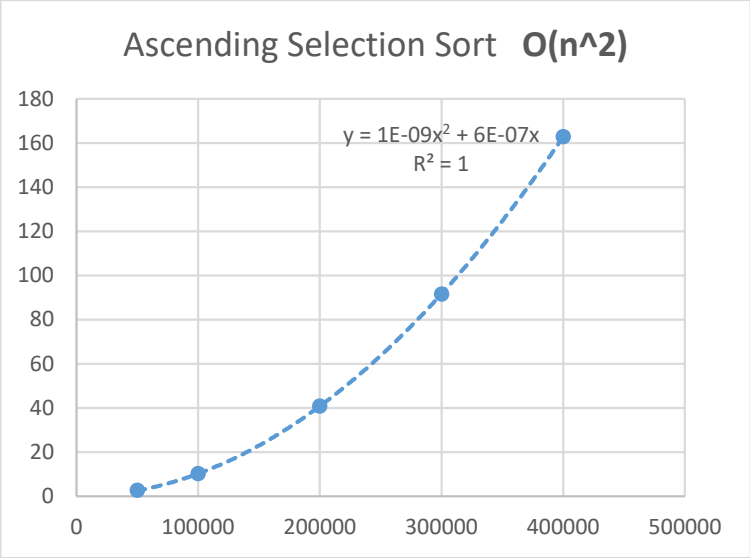
This column is N*Log_2(N)



Selection Sort

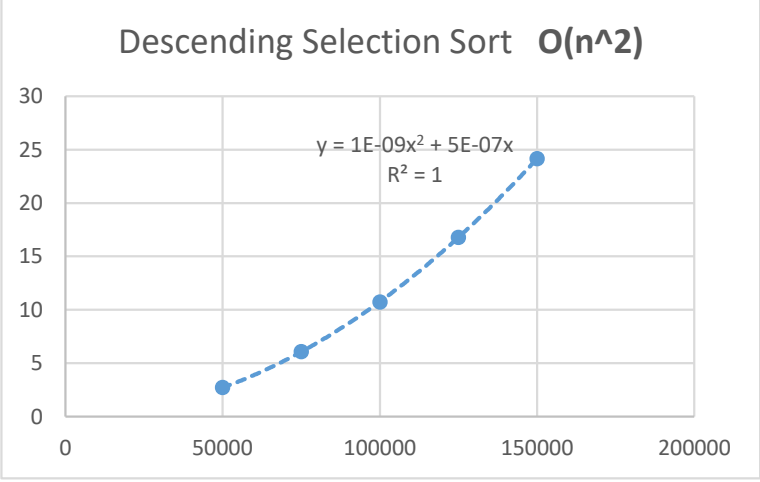
Ascending

50000	2.66838667
100000	10.2062
200000	40.8395
300000	91.5442
400000	162.941



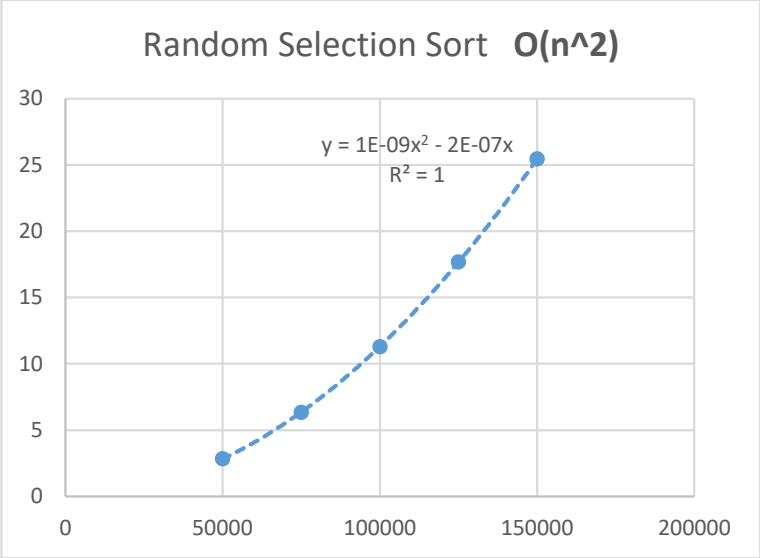
Descending

50000	2.71222
75000	6.06364333
100000	10.7381333
125000	16.7816333
150000	24.1632



Random

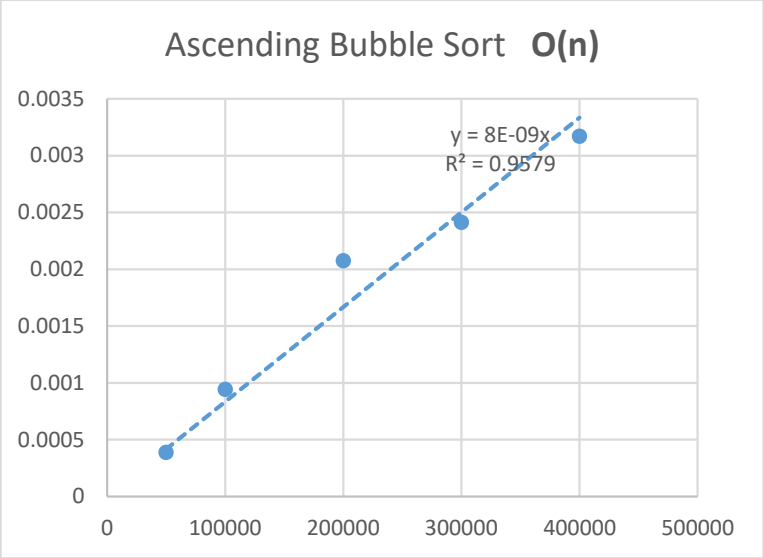
50000	2.83828333
75000	6.33717333
100000	11.3006
125000	17.6776333
150000	25.4523333



Bubble Sort

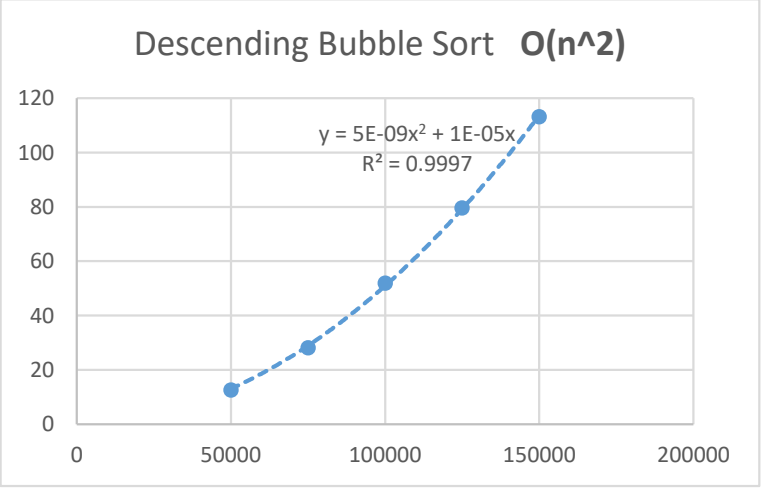
Ascending

50000	0.000388
100000	0.000942
200000	0.002075
300000	0.00241167
400000	0.00317167



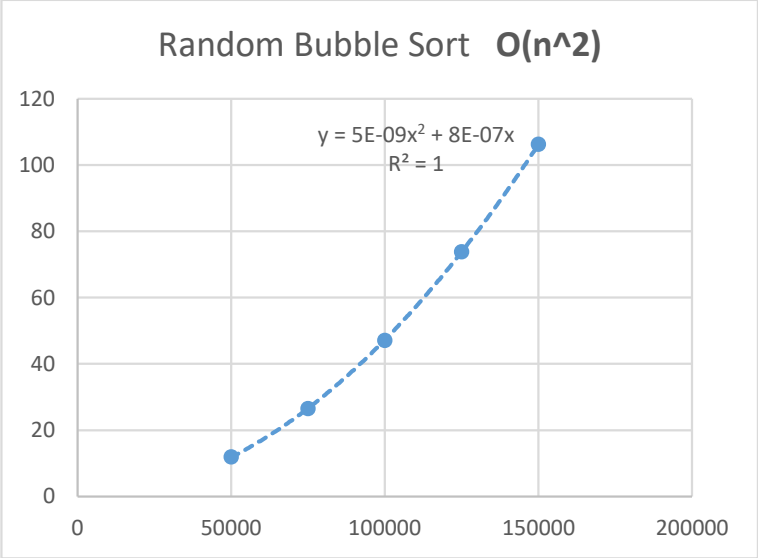
Descending

50000	12.5927333
75000	28.1534333
100000	51.9409
125000	79.6573667
150000	113.172333



Random

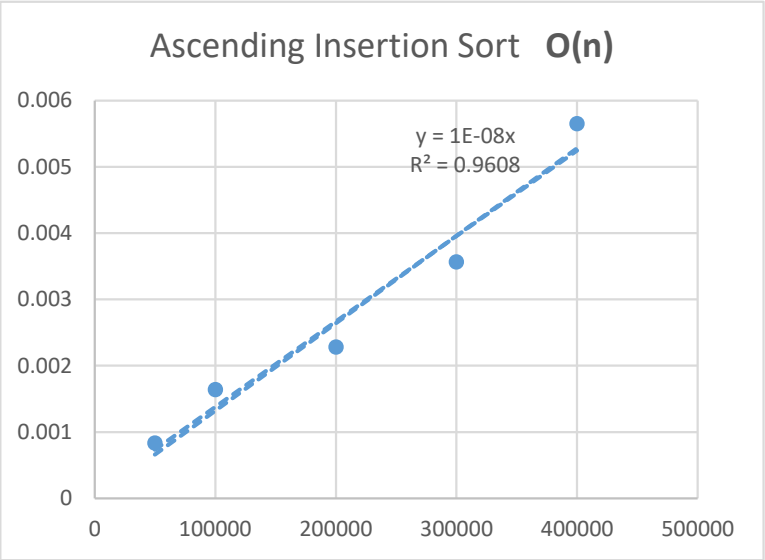
50000	11.9902667
75000	26.5357667
100000	47.0999
125000	73.8686
150000	106.252333



Insertion Sort

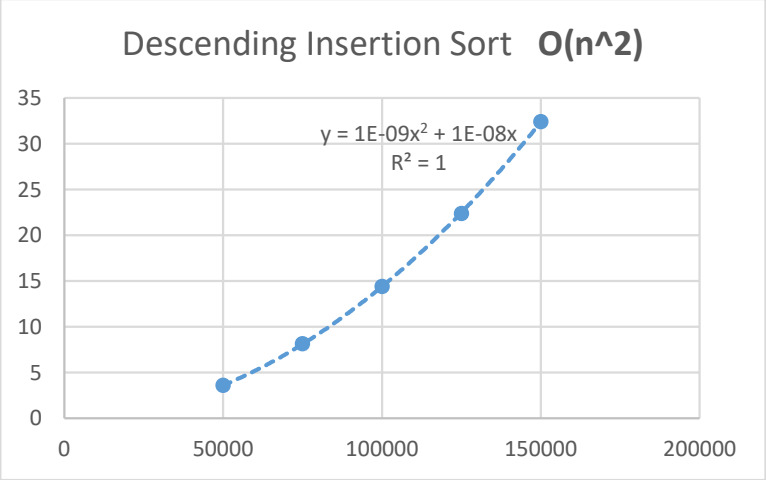
Ascending

50000	0.000836
100000	0.00164167
200000	0.00228133
300000	0.003567
400000	0.00565367



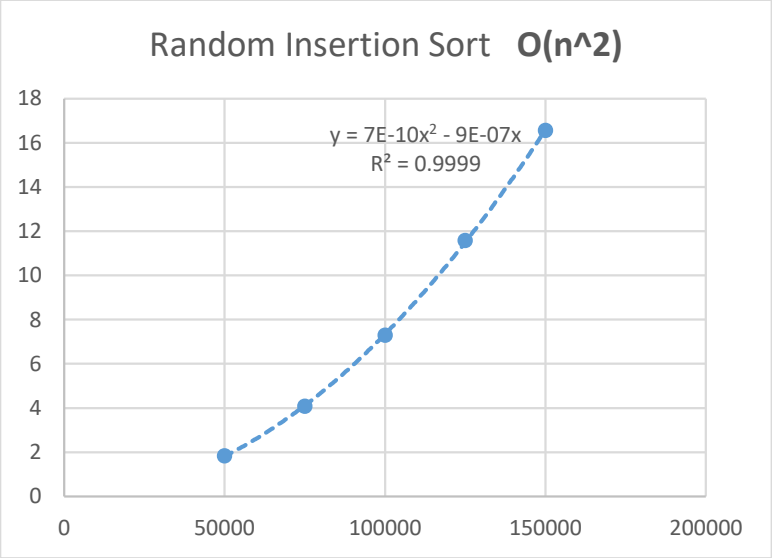
Descending

50000	3.60308667
75000	8.12510667
100000	14.4128667
125000	22.3604
150000	32.4234



Random

50000	1.84146333
75000	4.08526667
100000	7.29885333
125000	11.5788
150000	16.5613333



Results Note: All trials was run and tested on the same lab machine.

Insertion Sort should always have a growth rate of O(N^2). The graphs show this to be true with almost 100% certainty. Similarly, the "slow" sorts should also be O(N^2). They almost fit perfectly, with the worst having an R^2 of 0.9998. That's very good. I can say with an extremely high degree of confidence that all of the O(N^2) ones are correctly marked. That is, all of the Selection sorts, and the descending and randomly distributed insertion and bubble sorts.
In the case of the already sorted Insertion and Bubble sorts, all the algorithm would need to do is go through the array once and see that everything's already sorted. One pass through the array is O(N), so we should expect those to be O(N). Although the coefficient of determination (R^2) is only about 0.96 for both, it's still a good estimate, and considering the times have higher variance as a result of the sorts taking very little time, it's a very reasonable result.
If hundreds of trials were run on the ascending bubble/insertion sort, I would expect the variance to decrease, making our line of best fit more accurate.

Side note: Since all algorithms should take virtually zero seconds to sort an array of size zero, I forced the y-intercept to be 0 for all graphs. In most cases, this causes an almost negligable change in the line of best fit and coefficient of determination.

For the slow algorithms:

	Ascending	Descending	Random
Insertion	O(N)	O(N^2)	O(N^2)
Bubble	O(N)	O(N^2)	O(N^2)
Selection	O(N^2)	O(N^2)	O(N^2)

Predicted time to sort an array of size 10,000,000 (in seconds):

	Ascending	Descending	Random
Insertion	0.1	100000	69991
Bubble	0.08	500100	500008
Selection	100006	100005	99998

For the faster sorts, we should expect the ascending one to be linear, and the random ones to be O(NlogN). The descending ones are the worst-case scenario, so the worst case for quicksort is O(N^2) and the worst case for merge sort is still O(NlogN). In theory, it's easy to show those. However, since merge sort and quicksort are very efficient, even very large sizes, N in the millions get sorted in just a few seconds. This means every graph can be approximated relatively well by almost any type of function. So in order to find out which one it resembles most, we should look at the confidence coefficients. Every single one of them will be very close to 1, so it's mostly just a matter of trying them all and seeing which one is the best. And it does turn out that all of the functions most resemble their proven growth rate, even if the confidence coefficient only differs by a small amount. Even the O(NlogN) functions fit on a logarithmic graph better than a linear graph. Since N log N is not a commonly used function, to find its coefficient of determination I plotted N log N against time, and found the best linear fit for that graph. In every case, even the descending quicksort, the logarithmic graph fit better. So for the random and descending merge sorts, and the random quicksort, this was their best fit. The descending quicksort had R^2 = 0.9994, but a quadratic had R^2 = 0.9996, which, although a minor difference, means that at even larger values of N, it will behave more like O(N^2). It's also important to note that because the merge sort and quicksort do have more initial startup time compared to the slower sorts, I didn't force the graph to start at t=0 for n=0. Overall, my data supports each algorithm's growth rate, although the differences between calling some O(n) instead of O(nlogn) or O(n^2) can be very small, even at large values of n.
Side Note: The ascending merge sort modeled O(nlogn) better, making it have the largest O

For the fast algorithms:

	Ascending	Descending	Random
Merge	O(NlogN)	O(NlogN)	O(NlogN)
Quicksort	O(N)	O(N^2)	O(NlogN)

Predicted time to sort an array of size 10,000,000 (in seconds)

	Ascending	Descending	Random
Merge	0.13426	1.0221	2.12811
Quicksort	0.5345	0.97024	1.9028