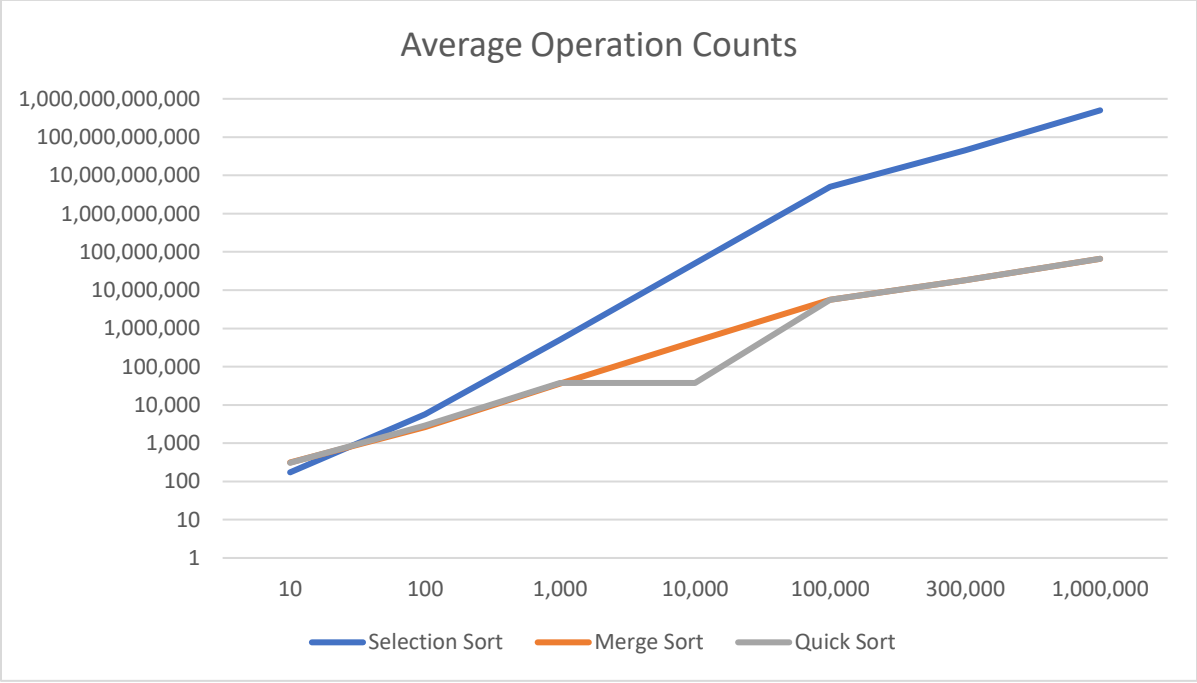
Nathan Young

To compare selection sort, merge sort, and quick sort, I wrote code to generate random arrays of integers in sizes ranging from 10, to 1,000,000. The arrays are random each time they are generated, and are re-generated every time a sorting algorithm is ran. For every algorithm, I implemented a counter to count the number of operations for every comparison, action and statements the algorithm performs.

After the algorithm completes a sort, the counter is reset to zero, and the next sort begins. All of the sorting algorithms take a variable-size array as the input and can be scaled indefinitely, though selection sort tends to be useless at sizes higher than 100,000.

- Selection sort, while giving the best results on arrays of less than 100, is by far the worst of the three algorithms tested, which should come as no surprise as its performance is $O(n^2)$.
- Merge sort yielded consistent results and was most effective for datasets under 10,000, though in larger quantities of data, it was barely worse than quick sort in terms of average performance. The difference is rather negligible on a modern computer with a fast processor. The performance is O(nlogn).
- Quick sort is heavily dependent on making "good" picks for its pivots, as it functions similarly to merge sort, its average performance is O(nlogn), though worst case it can be as bad as $O(n^2)$. However worst case usually can be considered statistically impossible. Of the three sorts, quick sort was the fastest to complete, though often had near identical operations as merge sort.

A detailed set of tables and graph follow:



Nathan Young

Selection Sort							
Trial	10	100	1,000	10,000	100,000	300,000	1,000,000
1	175	5,689	509,099	50,112,968	5,001,330,586	45,004,233,099	500,014,443,224
2	174	5,665	508,827	50,113,831	5,001,335,311	45,004,215,190	500,014,423,780
3	172	5,646	508,989	50,112,547	5,001,335,795	45,004,229,981	500,014,415,689
4	171	5,672	509,045	50,112,384	5,001,334,142	45,004,227,752	500,014,426,412
5	173	5,696	509,142	50,111,674	5,001,338,481	45,004,183,875	500,014,434,249
Average	173	5,674	509,020	50,112,681	5,001,334,863	45,004,217,979	500,014,428,671
Merge Sort							
Trial	10	100	1,000	10,000	100,000	300,000	1,000,000
1	314	2,611	35,923	460,843	5,606,779	18,227,131	65,854,267
2	314	2,611	35,923	460,843	5,606,779	18,227,131	65,854,267
3	314	2,611	35,923	460,843	5,606,779	18,227,131	65,854,267
4	314	2,611	35,923	460,843	5,606,779	18,227,131	65,854,267
5	314	2,611	35,923	460,843	5,606,779	18,227,131	65,854,267
Average	314	2,611	35,923	460,843	5,606,779	18,227,131	65,854,267
Quick Sort							
Trial	10	100	1,000	10,000	100,000	300,000	1,000,000
1	300	2,954	37,757	37,615	5,580,102	18,471,866	66,551,204
2	273	2,855	37,714	37,899	5,681,178	18,261,059	66,140,010
3	330	2,811	37,922	37,937	5,573,891	18,062,113	65,257,007
4	287	2,859	37,499	37,836	5,672,247	18,178,905	67,223,963
5	343	2,873	36,467	37,481	5,682,610	18,413,646	65,087,200
Average	307	2,870	37,472	37,754	5,638,006	18,277,518	66,051,877
Averages							
Sort	10	100	1,000	10,000	100,000	300,000	1,000,000
Selection Sort	173	5,674	509,020	50,112,681	5,001,334,863	45,004,217,979	500,014,428,671
Merge Sort	314	2,611	35,923	460,843	5,606,779	18,227,131	65,854,267
Quick Sort	307	2,870	37,472	37,754	5,638,006	18,277,518	66,051,877