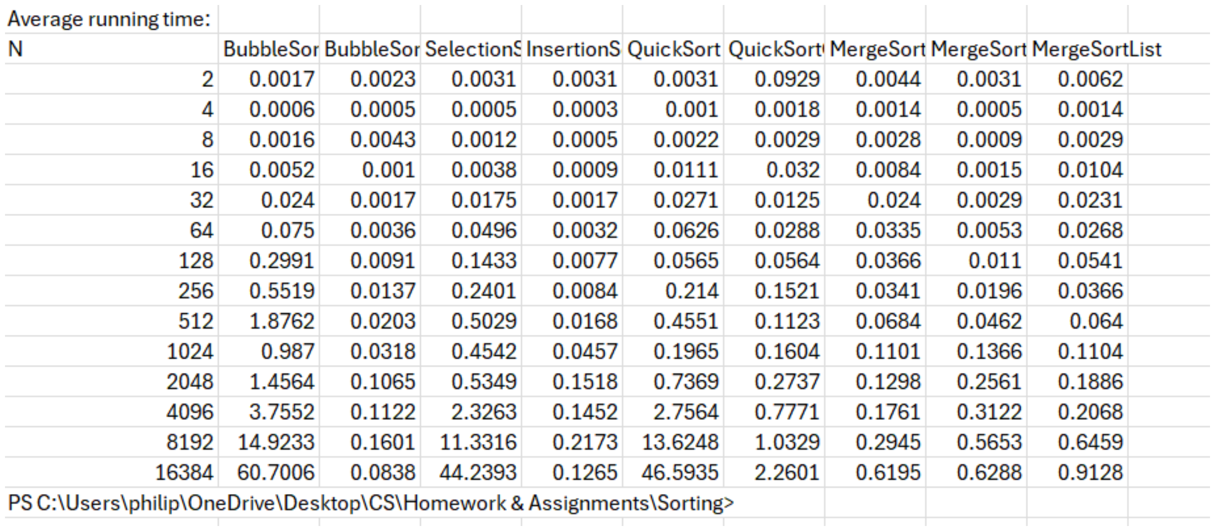
/*

THIS CODE WAS MY OWN WORK, IT WAS WRITTEN WITHOUT CONSULTING

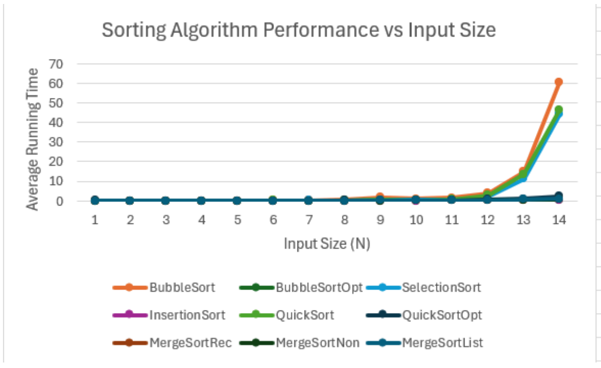
CODE WRITTEN BY OTHER STUDENTS OR COPIED FROM ONLINE RESOURCES.
Philip Cardozo

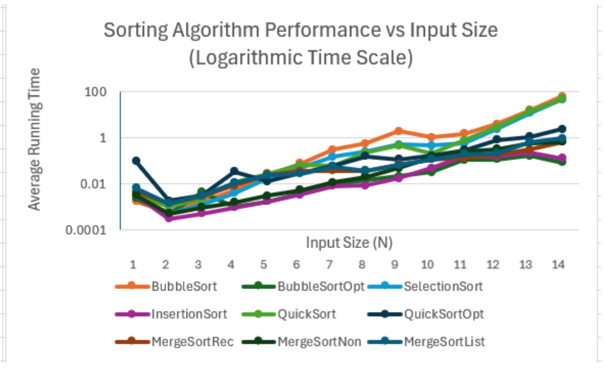
*/



Average running time:									
N	BubbleSor	BubbleSor	SelectionS	InsertionS	QuickSort	QuickSort	MergeSort	MergeSort	MergeSortList
2	0.0017	0.0023	0.0031	0.0031	0.0031	0.0929	0.0044	0.0031	0.0062
4	0.0006	0.0005	0.0005	0.0003	0.001	0.0018	0.0014	0.0005	0.0014
8	0.0016	0.0043	0.0012	0.0005	0.0022	0.0029	0.0028	0.0009	0.0029
16	0.0052	0.001	0.0038	0.0009	0.0111	0.032	0.0084	0.0015	0.0104
32	0.024	0.0017	0.0175	0.0017	0.0271	0.0125	0.024	0.0029	0.0231
64	0.075	0.0036	0.0496	0.0032	0.0626	0.0288	0.0335	0.0053	0.0268
128	0.2991	0.0091	0.1433	0.0077	0.0565	0.0564	0.0366	0.011	0.0541
256	0.5519	0.0137	0.2401	0.0084	0.214	0.1521	0.0341	0.0196	0.0366
512	1.8762	0.0203	0.5029	0.0168	0.4551	0.1123	0.0684	0.0462	0.064
1024	0.987	0.0318	0.4542	0.0457	0.1965	0.1604	0.1101	0.1366	0.1104
2048	1.4564	0.1065	0.5349	0.1518	0.7369	0.2737	0.1298	0.2561	0.1886
4096	3.7552	0.1122	2.3263	0.1452	2.7564	0.7771	0.1761	0.3122	0.2068
8192	14.9233	0.1601	11.3316	0.2173	13.6248	1.0329	0.2945	0.5653	0.6459
16384	60.7006	0.0838	44.2393	0.1265	46.5935	2.2601	0.6195	0.6288	0.9128

```
PS C:\Users\philip\OneDrive\Desktop\CS\Homework & Assignments\Sorting>
```





The optimized version of QuickSort consistently outperforms the standard QuickSort, especially on sorted arrays where the unoptimized version degrades to *O* (*n* 2) O(n 2). This highlights how shuffling and insertion sort cutoffs improve robustness. BubbleSortOptimized and InsertionSort both perform nearly linearly on sorted inputs, while naive BubbleSort and SelectionSort still do unnecessary *O* (*n* 2) O(n 2) work. For large random arrays, *O* (*n* 2) O(n 2) algorithms quickly become impractical, while MergeSort and QuickSort variants scale efficiently due to their *O* (*n* log *n*) O(nlogn) complexity. The recursive and non-recursive MergeSort implementations show nearly identical performance, with the non-recursive version slightly faster at large sizes. Merge sort also performs well on linked lists, making it a strong choice for pointer-based data structures. Overall, the results show that algorithm choice and input structure significantly impact performance.