DSA 5005 - Project 6
Sorting Algorithm Analysis
Jacob Flynn

Introduction:
When looking to sort a list of numbers, there is a multitude of sorting algorithms that exist to do so; each with their own functionality and ranges of complexity. Two sorting algorithms will be presented in this analysis, the Bubble Sort and Shell Sorting techniques with an increasing number of elements (n) ranging from 1,000 to 30,000. Each technique is in order $O(n^2)$ in the worst-case scenario, when the numbers are in reverse order, and optimally O(n) when the list is already sorted. The objective of this project is to determine the most efficient sorting method by calculating quality metrics, Inversions and Chebychev's Distance, for differing numbers of randomly assorted elements.

Analysis:
As mentioned previously, the number of inversions and Chebyshev's Distance will be used for each quality metric. The number of inversions is calculated by counting the number of elements to the right that are less than the current element, this number indicates how far a list is from being sorted. Chebyshev's Distance is calculated by taking the sum and absolute value of how many index positions the current element is away from its sorted position. This number also indicates the minimum distance the elements are from being sorted.

In the following graphs, the blue lines will indicate the Bubble Sort algorithm while red will indicate the Shell Sorting algorithm. The graphs are semi-logarithmic across the x-axis as this reveals the best distribution of points. The graphs are also calculated to the worse-case, $O(n^2)$, when every element has been sorted.

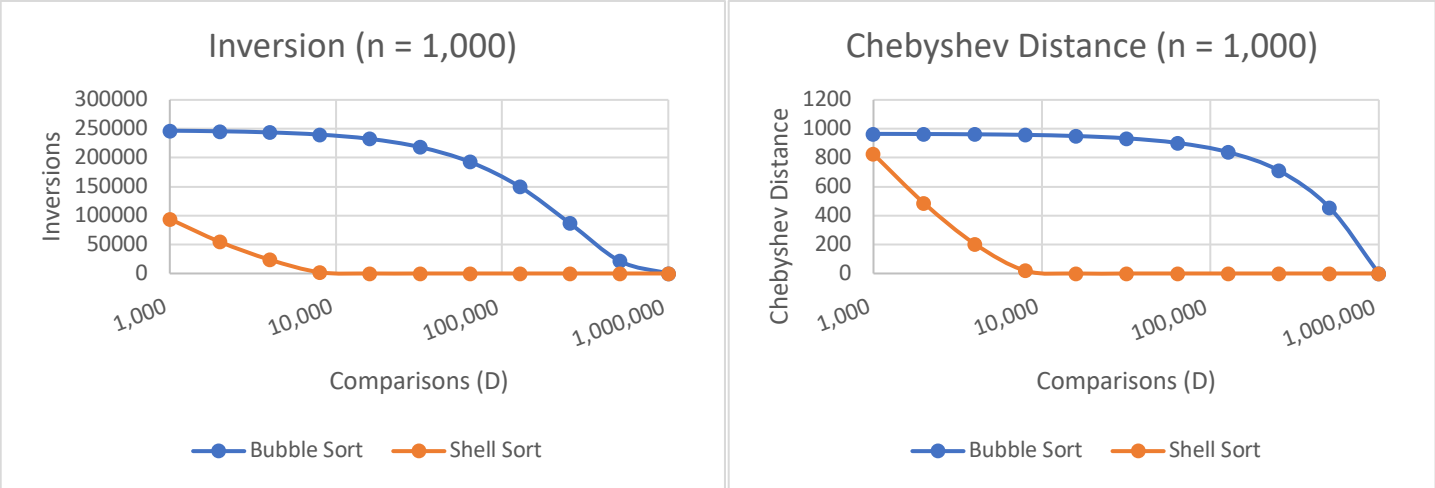


Fig. 1- The number of inversions and Chebyshev's Distance for 1000 elements

Seen above in **Fig. 1**, it displays the number of inversions and Chebyshev's Distance over the number of allowed comparisons (D) for 1,000 randomly assigned elements. Shell Sort outperforms the Bubble Sorting algorithm in both quality metrics by less than half of the comparisons. This tells that even though Bubble and Shell Sort have the same complexity, Shell is able to sort the elements with less than half the number of comparisons and provide a more accurate initial sort.

This will be true for all cases seen below for n equal to 5,000, 10,000, 20,000, and 30,000. The tables used to generate each graph can be found in the **Appendix** with their respective table number.

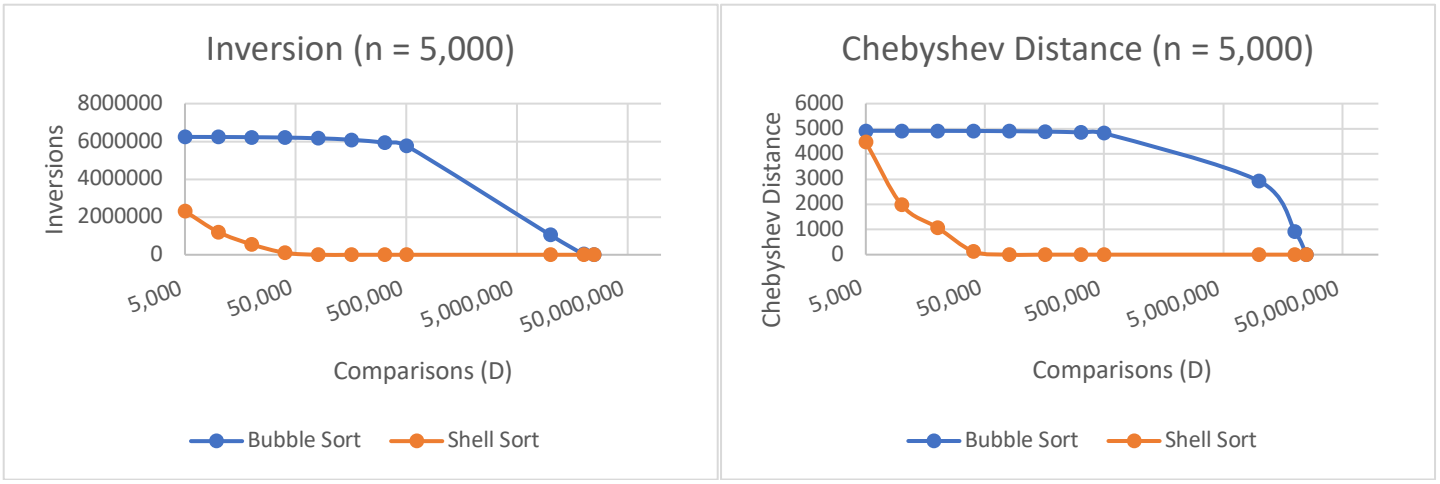


Fig. 2- The number of inversions and Chebyshev's Distance for 5,000 elements

In **Fig. 2** and the following graphs, there is a sharp decline seen in both the inversion and Chebyshev Distance during the Bubble Sort algorithm at 500,000 comparisons. This may be due to the Bubble Sorts ability to switch element positions from left to right, one place at a time, during the algorithm. For the case of 5,000 elements, after 500,000 comparisons of the total 50,000,000, a majority of the elements will have already been sorted, thus decreasing the number of inversions and the Chebyshev's Distance. This will be true for the following number of elements as seen in the graphs as well.

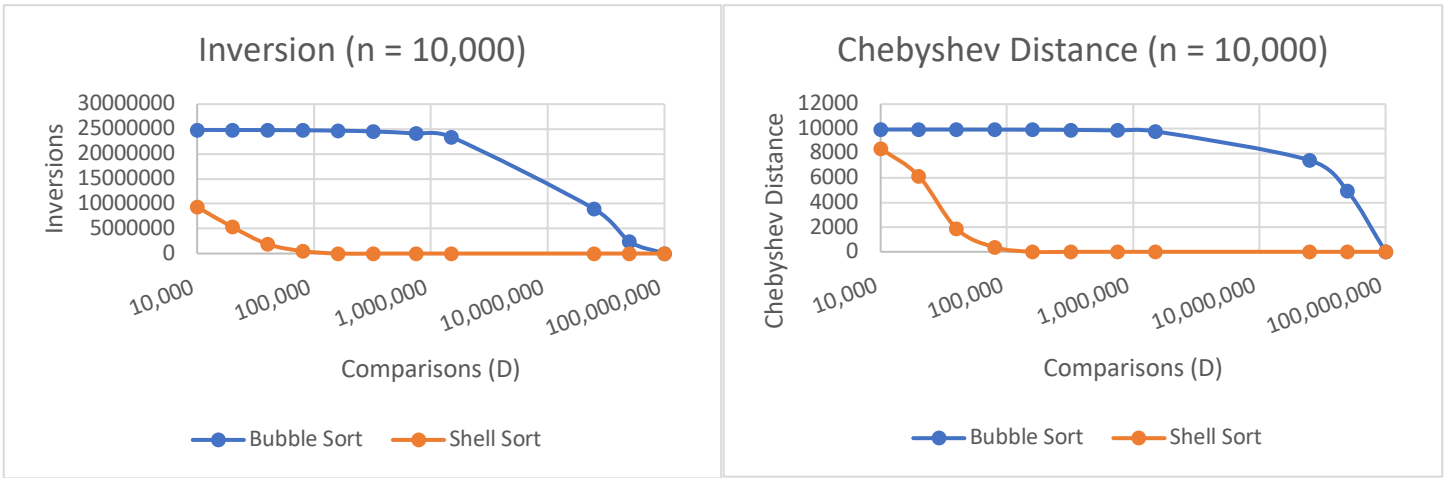


Fig. 3- The number of inversions and Chebyshev's Distance for 10,000 elements

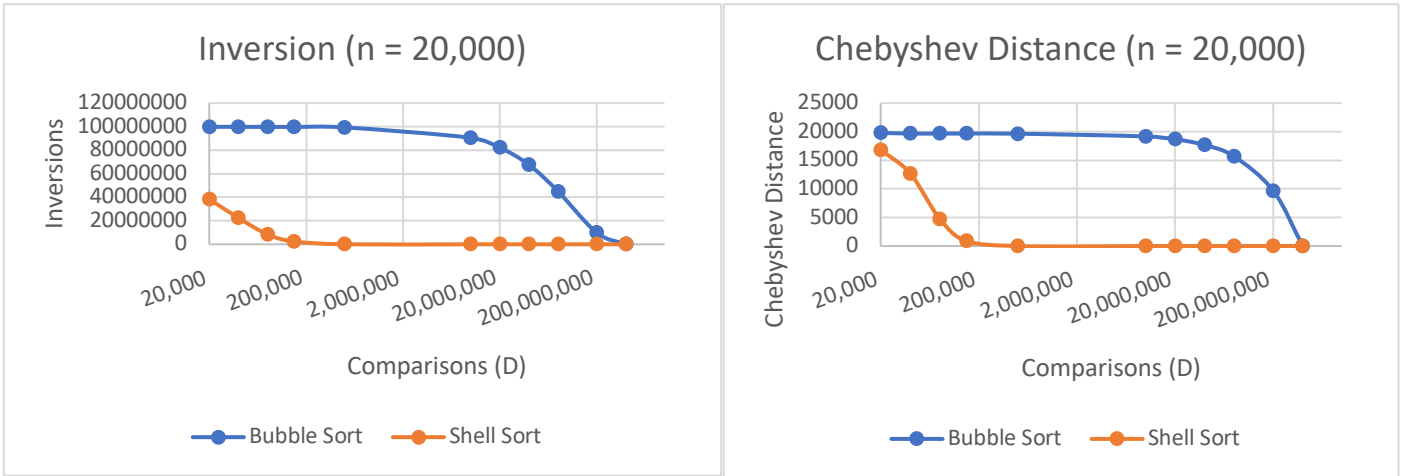
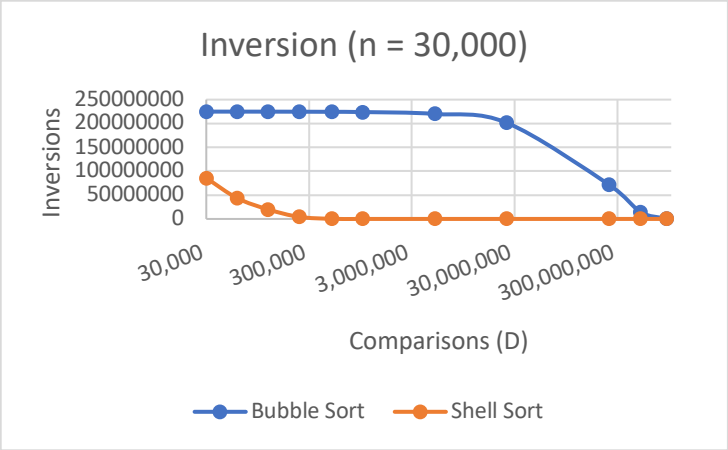


Fig. 4- The number of inversions and Chebyshev's Distance for 20,000 elements



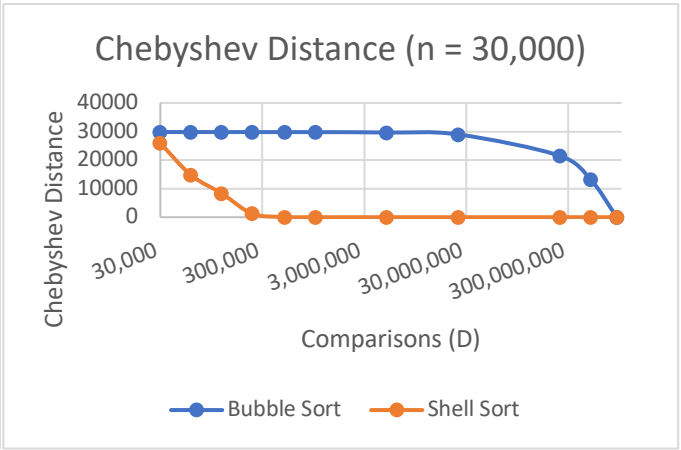


Fig. 5- The number of inversions and Chebyshev's Distance for 30,000 elements

Conclusion:

Concluding the analysis, it can be shown that Shell Sort is a much faster sorting algorithm than Bubble Sort as seen by the number of inversions and Chebyshev's Distance decrease dramatically faster across those quality metrics. Though after roughly 100x the number of comparisons to elements, the Bubble Sorting algorithm decreases significantly across both metrics. If the number of elements and allowed comparisons is few, either method would work without noticing decrease in sorting time, but as the number of elements increase, Shell Sort is recommended as it can reduce sorting time significantly.

Appendix:

n = 1000	n2 =	1,000,000		
D	Bub Inv	Bub Chev	Shell Inv	Shell Chev
1,000	246743	965	94050	825
2,000	245755	964	54700	485
4,000	243791	962	23778	204
8,000	239965	958	2064	19
16,000	232558	950	0	0
32,000	218398	934	0	0
64,000	192801	902	0	0
128,000	149710	838	0	0
256,000	86627	710	0	0
512,000	21322	453	0	0
1,000,000	0	0	0	0

Table A1 – Numbers calculated to generate Figure 1

n = 5000	n2 =	25,000,000		
D	Bub Inv	Bub Chev	Shell Inv	Shell Chev
5,000	6237663	4920	2305254	4471
10,000	6232680	4919	1204770	1973
20,000	6222741	4917	553652	1071
40,000	6202940	4913	106448	129
80,000	6163647	4905	0	0
160,000	6086097	4889	0	0
320,000	5934809	4857	0	0
500,000	5770634	4821	0	0
10,000,000	1046055	2921	0	0
20,000,000	28230	920	0	0
25,000,000	0	0	0	0

Table A2 – Numbers calculated to generate Figure 2

n = 10000	n2 =	100,000,000		
D	Bub Inv	Bub Chev	Shell Inv	Shell Chev
10,000	24816306	9936	9354525	8402
20,000	24806325	9935	5331645	6166
40,000	24786378	9933	1892599	1874
80,000	24746604	9929	499651	372
160,000	24667441	9921	0	0
320,000	24510420	9905	0	0
750,000	24095524	9862	0	0
1,500,000	23393422	9787	0	0
25,000,000	8954743	7437	0	0
50,000,000	2419259	4936	0	0
100,000,000	0	0	0	0

Table A3 – Numbers calculated to generate Figure 3

n = 20000	n2 =	400,000,000		
D	Bub Inv	Bub Chev	Shell Inv	Shell Chev
20,000	99846472	19790	38437431	16851
40,000	99826490	19674	22379023	12705
80,000	99786560	19672	8446227	4764
150,000	99716798	19669	2396891	871
500,000	99369112	19651	0	0
10,000,000	90518172	19176	0	0
20,000,000	82130382	18676	0	0
40,000,000	67511215	17676	0	0
80,000,000	44804757	15676	0	0
200,000,000	9809650	9675	0	0
400,000,000	0	0	0	0

Table A4 – Numbers calculated to generate Figure 4

n = 30000	n2 =	900,000,000		
D	Bub Inv	Bub Chev	Shell Inv	Shell Chev
30,000	224857543	29809	85565306	25904
60,000	224827564	29808	43514813	14753
120,000	224767633	29806	19947589	8306
240,000	224647903	29802	4322087	1236
500,000	224388857	29794	245117	74
1,000,000	223892043	29777	0	0
5,000,000	219981681	29644	0	0
25,000,000	201635804	28977	0	0
250,000,000	71292935	21477	0	0
500,000,000	14665736	13143	0	0
900,000,000	0	0	0	0

Table A5 – Numbers calculated to generate Figure 5