Report - Project 6

In this report we will discuss two metrics, Inversion and Chebyshev distance to analyze the two sorting algorithms, Bubble & Shell sort. We will do it by varying number of elements in an array and limiting number of comparisons while sorting in these two algorithms. Below are five cases which we have considered to analyze.

1. Number of elements(n) are 1000

First we will check the table we've got after testing 1000 elements by changing D value.

N - 1000	Bubble Sort		Shell Sort	
D(Comparisions)	Inversions	Chebyshev	Inversions	Chebyshev
700	254121	950	114416	781
1500	253329	930	67270	433
5000	249894	904	17558	221
8000	246997	901	2575	19
17000	238598	892	0	0
40000	218655	869	0	0
85000	183996	824	0	0
180000	125716	729	0	0
370000	52912	539	0	0
520000	22162	388	0	0
790000	1315	118	0	0
850000	315	58	0	0
920000	0	0	0	0

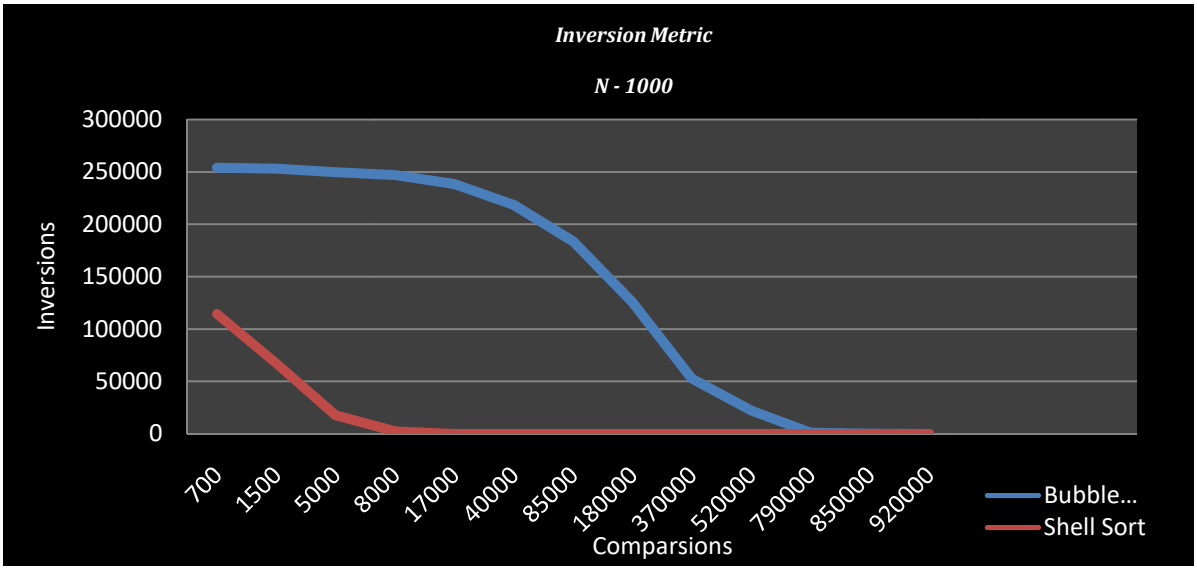

Fig.1 Inversion metric for bubble & shell sort for 1000 elements

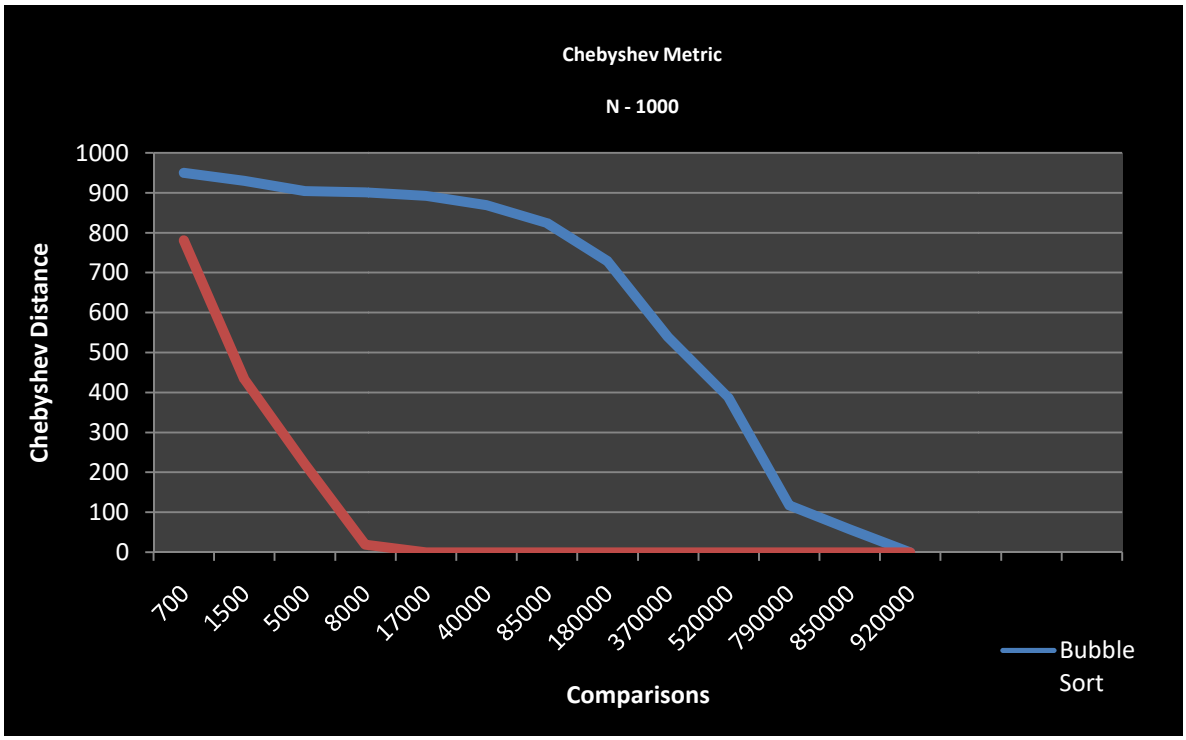

Fig.2 Chebyshev distance metric for bubble & shell sort for 1000 elements

Fig1 & Fig2 shows the line chart of two metrics applied on bubble & shell sort with 1000 elements and with varying D value until we see 0 inversions and 0 chebyshev distance. In this we have seen bubble sort doesn't show sharp decline in these two metrics with increase in comparisons like shell sort did. Bubble sort require a lot more comparisons to get the array sorted fully when compared to shell sort. We came to conclusion that bubble sort needs **450% more** comparisons than shell sort to sort an array fully when number of elements are around 1000.

2. Number of elements(n) are 5000

N - 5000	Bubble Sort		Shell Sort	
D(Comparisions)	Inversions	Chebyshev	Inversions	Chebyshev
500	6227628	4982	5673407	4982
1500	6226628	4982	4501758	4982
5000	6223130	4981	2380063	4311
15000	6213171	4979	764220	1224
45000	6183433	4973	63174	125
60000	6168683	4970	6888	13
90000	6139367	4964	0	0
180000	6052489	4946	0	0
520000	5738343	4878	0	0
5500000	2557004	3882	0	0
8800000	1376537	3222	0	0

13200000	493386	2341	0	0
22700000	3052	441	0	0
32700000	0	0	0	0

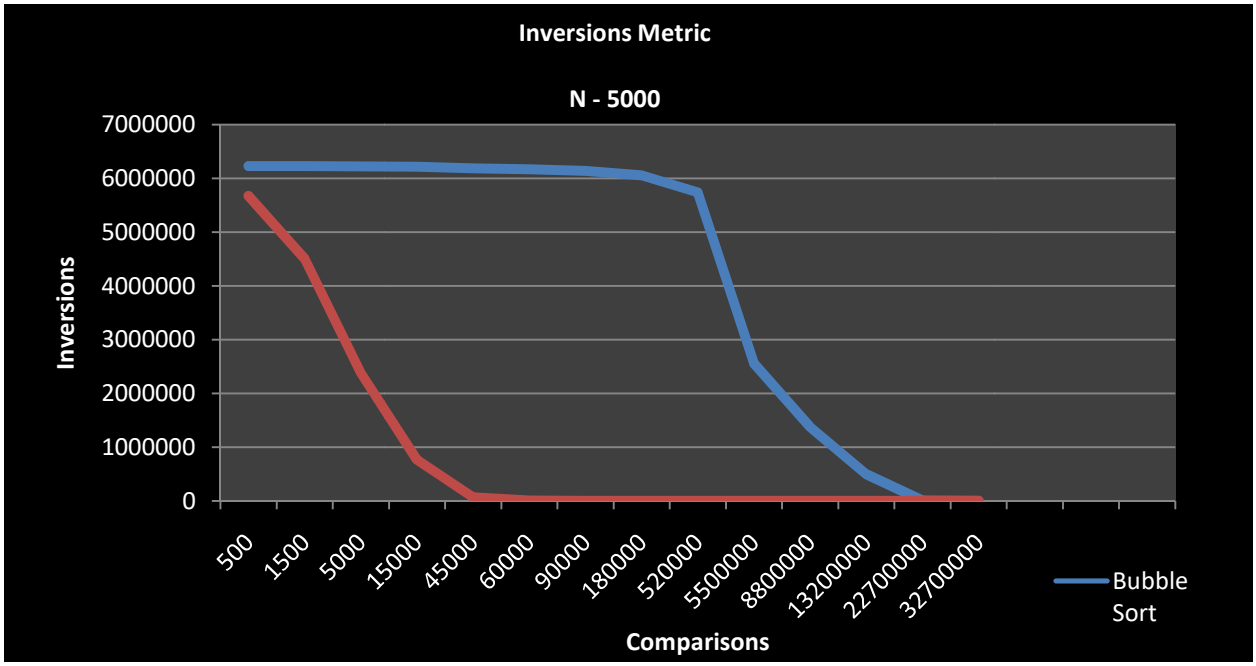

Fig.3 Inversion metric for bubble & shell sort for 5000 elements

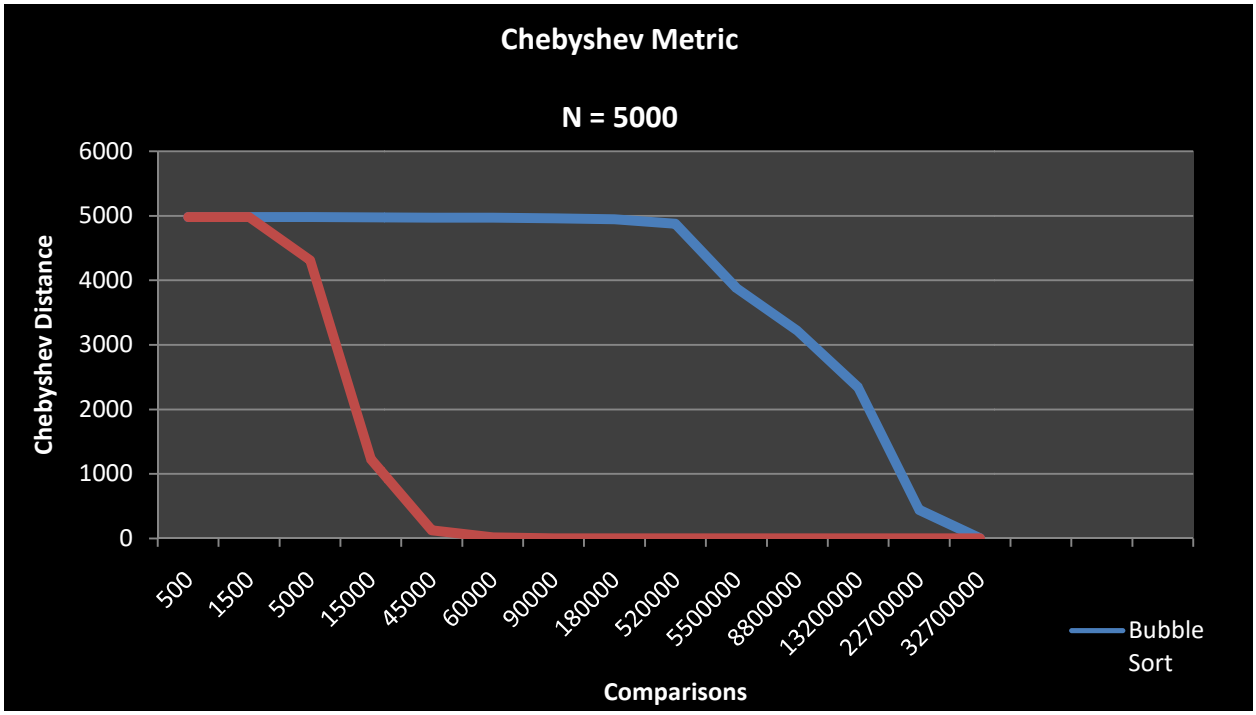

Fig.4 Chebyshev distance metric for bubble & shell sort for 5000 elements

Fig3 & Fig4 shows the line chart of two metrics applied on bubble & shell sort with **5000** elements and with varying D value until we see 0 inversions and 0 chebyshev distance. Shell sort quickly sort more elements when number of comparisons increased but bubble sort doesn't do so until a certain point is reached. For 5000 elements bubble show growth in sorting after about **50,000** comparisons but shell sort quickly sort elements as we increase comparisons and sort all elements with about **60000** comparisons. It means when elements are approximately **5,000** bubble sort needs **330000** comparisons which is again **450% more** compared to shell sort.

3. Number of elements(n) are 10000

N - 10000	Bubble Sort		Shell Sort	
D(Comparisions)	Inversions	Chebyshev	Inversions	Chebyshev
5000	24681068	9903	13186553	8630
25000	24661118	9901	4325793	5577
65000	24621335	9897	872856	757
150000	24537109	9888	17456	13
450000	24243828	9858	0	0
1500000	24243828	9858	0	0
6500000	19143254	9253	0	0
12500000	15095275	8653	0	0
25800000	8592360	7323	0	0
35000000	5561815	6403	0	0
85000000	66831	1402	0	0
95000000	2049	402	0	0
98000000	122	102	0	0
120000000	0	0	0	0

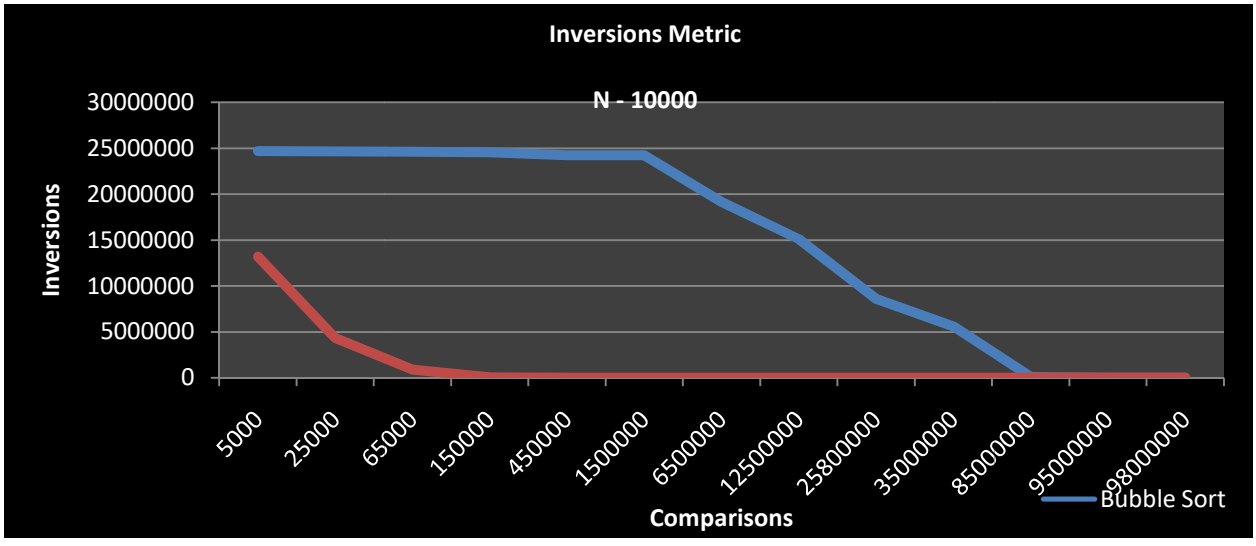

Fig.5 Inversion metric for bubble & shell sort for 10000 elements

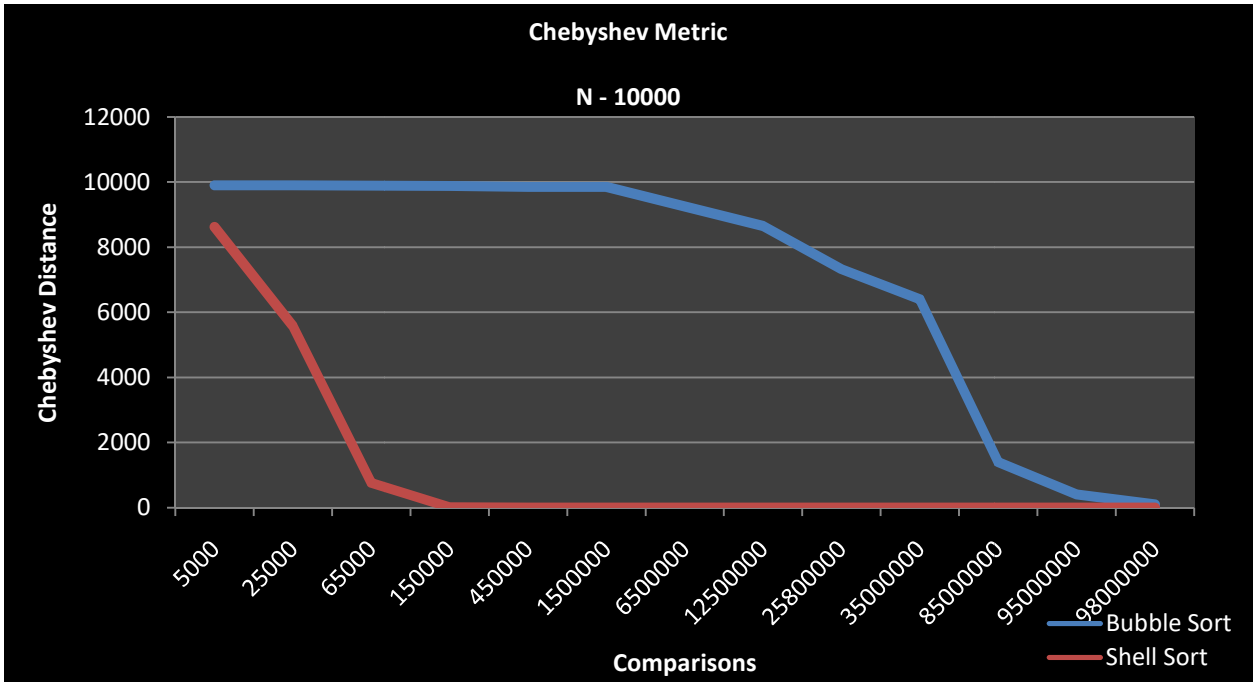

Fig.6 ChebyShev metric for bubble & shell sort for 10000 elements

Fig.4 & Fig.5 shows the line chart of two metrics applied on bubble & shell sort with **10000** elements and with varying D value until we see 0 inversions and 0 chebyshev distances. Here we have noticed that as we are increasing number of elements bubble sorts is lagging far behind shell sort. For **10000** elements bubble sort require **700% more** comparisons than shell sort. As the number of elements is increasing bubble sort's performance is getting worst.

4. When number of elements(n) are 20000

N - 20000	Bubble Sort		Shell Sort	
D(Comparisions)	Inversions	Chebyshev	Inversions	Chebyshev
15000	100500973	19812	45725110	18393
45000	100471018	19810	19269990	11257
90000	100426099	19808	6296504	4118
180000	100336343	19803	1796835	961
360000	100157345	19794	51178	24
520000	99998767	19786	0	0
1024000	99502248	19761	0	0
10240000	90985474	19300	0	0
102400000	36702574	14692	0	0
155000000	20258628	12062	0	0
220000000	7847728	8811	0	0
240000000	5451686	7811	0	0

290000000	1687028	5311	0	0
320000000	629935	3811	0	0
340000000	253988	2811	0	0
390000000	1380	311	0	0
420000000	0	0	0	0

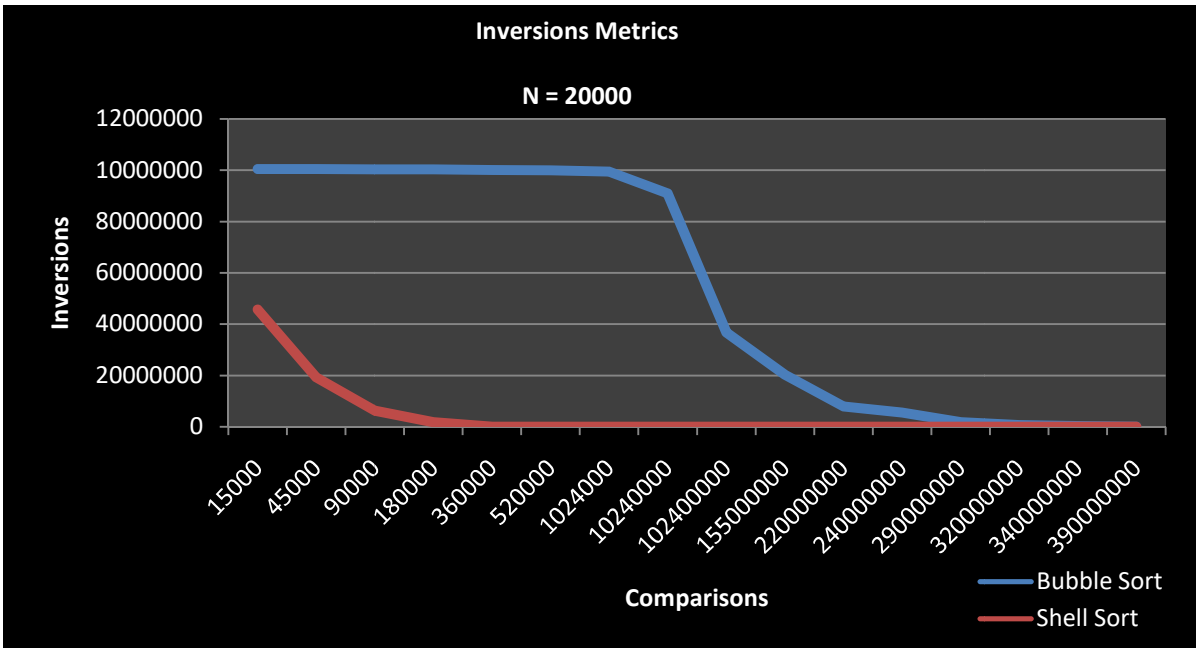

Fig.7 Inversion metric for bubble & shell sort for 20000 elements

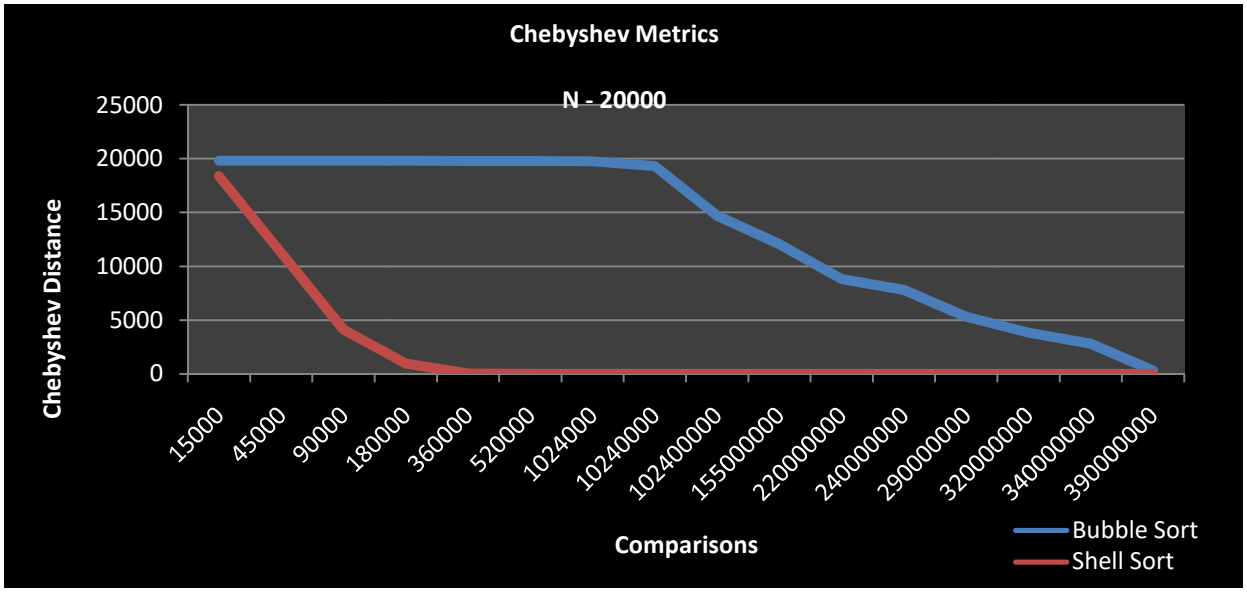

Fig.8 ChebyShev metric for bubble & shell sort for 20000 elements

Fig.7 & Fig.8 shows the line chart of two metrics applied on bubble & shell sort with **20000** elements and with varying D value until we see 0 inversions and 0 chebyshev distances. Here we have noticed that as we are increasing number of elements bubble sorts is lagging far behind shell sort. Bubble sort doesn't show any improvement in sort until the number of comparison reached **10000000**. For **20000** elements bubble sort require **1000% more** comparisons than shell sort. As the number of elements is increasing bubble sort's performance is getting worst.

5. When number of elements(n) are 30000

N - 30000	Bubble Sort		Shell Sort	
D(Comparisions)	Inversions	Chebyshev	Inversions	Chebyshev
20000	226437668	29717	109190316	26946
80000	226377705	29601	30110226	9254
150000	226307780	29598	11273018	5271
300000	226158082	29593	2972726	712
450000	226008588	29588	1128790	567
850000	225610967	29575	0	0
1850000	224622605	29542	0	0
18500000	208982014	28987	0	0
185000000	99456352	23436	0	0
485000000	14802113	13436	0	0
685000000	1755122	6769	0	0
785000000	241630	3435	0	0
820000000	85692	2268	0	0
890000000	0	0	0	0

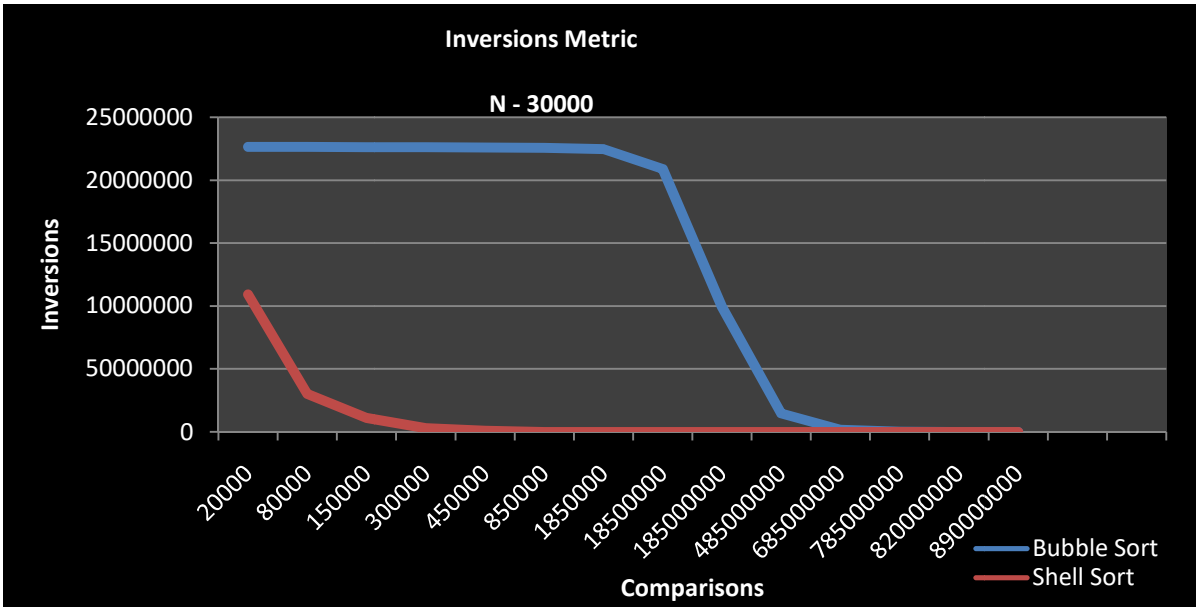

Fig.9 Inversion metric for bubble & shell sort for 30000 elements

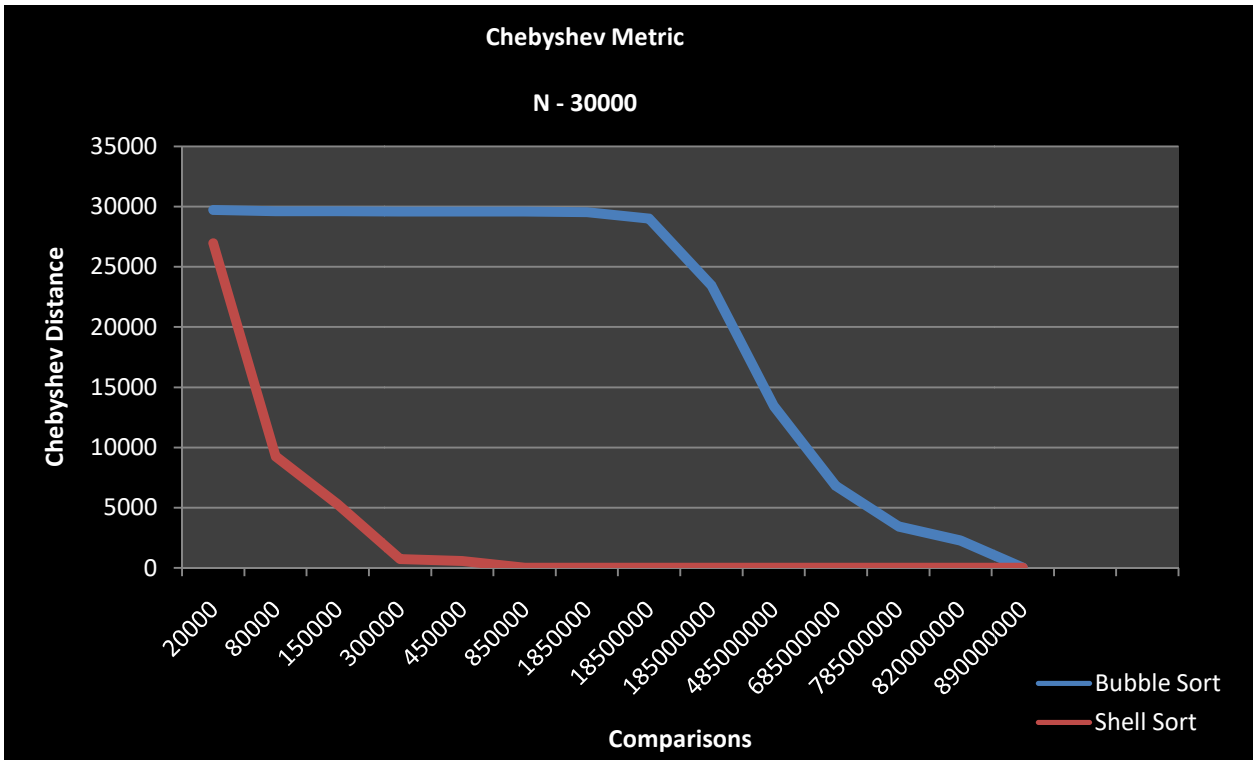

Fig.10 ChebyShev metric for bubble & shell sort for 30000 elements

Fig.9 & Fig.10 shows the line chart of two metrics applied on bubble & shell sort with **30000** elements and with varying D value until we see 0 inversions and 0 chebyshev distances. We have noticed that as the numbers of elements are increasing bubble sort's performance is getting worst as it is requiring a lot more comparisons to get the array sorted. When elements are **30000** we have noticed that bubble sort needs **1000% more** comparisons than shell sort.

Conclusion:

After checking Performance metrics we can confirm that shell sort is better that bubble sort because it sorts the entire array with a lot less comparisons than bubble sort. Bubble sorts performance decreases with the increase in number of elements.

We have also witnessed, in our 5 cases that, for

1. **1000 & 5000** elements array bubble sort making **450% more** comparisons than shell sort.
2. **10000** elements array bubble sort making **700% more** comparisons than shell sort.
3. **1000 & 5000** elements array bubble sort making **1000% more** comparisons than shell sort.

With these points we can say that shell sorts performance is better than bubble sort as it sorts elements with very less comparisons.