ISSN:2229-6093

Avi Sorting Network

Avinash Bansal
Assistant Professor (CSE)
GNIT, Mullana.
Ambala (Haryana), India
avinashbansal26@gmail.com

***Abstract*— Sorting network is an abstract mathematical model which can be used as a multiple-input, multiple-output switching network to sort the data in ascending or descending order [1]. Sorting has been one of the most critical applications on parallel computing machines. Many classic textbooks on algorithms like Thomas H. Cormen, therefore consider this problem in great detail and list many sorting network for this purpose [2]. There are many sorting algorithms as the Bubble / Insertion sorter, Odd-Even sorter, Sort the data in O($\log_2 n)^2$ time complexity and some other sorter have O(n^2) as time complexity, where n is the number of elements. In this paper we propose a sorting network called "Avi Sorter" having time complexity O(n $\log_2$ n) which is based on just similar to bubble sort algorithm. This sorting network provides the easy way to understand and manipulate the concept of sorting network.**

Keywords: Sorting, Parallel, Network, Avi and Comparator.

I. Introduction

Parallel model of computation uses comparison networks. Roughly speaking, a comparison network is an algorithm that allows multiple comparisons to be made concurrently [3]. This process requires comparator. A comparator is a device with two inputs and two outputs. Comparator is shown in Fig 1. Fig1 (a) shows general representation of comparator whereas Fig 1 (b) depicts the pictorial representation of Fig1 (a).

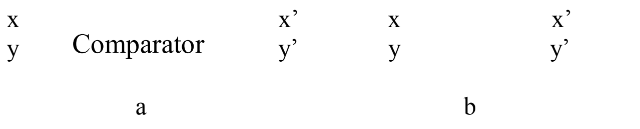

Fig.1 Comparator

In Fig 1, x and y represent the input data element and x' and y' are output data element where x' = min (x, y) and y' = max (x, y). Each comparators operates O(1) time [3]. Any sorting network on n inputs has depth at least $\log_2$n and number of comparators in any sorting network is at least Ω (n $\log_2$ n) [3].

There are many sorting networks such as linear search based network have O (n^2) time complexity, Odd-Even sorting network and merge sorting network, both network sort the data in O ($\log_2^2$ n) time. Similarly other sorting network like AKS network also sort the data in O ($\log_2$ n) time, which is the minimum time complexity of the sorting network till now [4]. Unfortunately, the constant hidden by the O-notation in AKS sorting network are quite large (many, many thousands), and thus it cannot be consider practically [4]. Time complexity of different sorting networks represent in Table I.

TABLE I
SORTING NETWOK COMPLEXITY

S. No.	Sorting Network	Time Complexity	Order
1.	Linear Search based Sorting Network	O (n^2) [6]	
2.	Even-Odd Network Merge Network Aviva Sorting Network	O ($\log_2^2$ n) [7]	
3.	AKS Network	O ($\log_2$ n)	

II. Proposed Work

There are many sorting networks which sort the data in increasing or decreasing order according to the type of comparator. This network is slightly similar to bubble sort algorithm, in which the heaviest element settles down at the bottom of the network and lighter elements move up. Avi sorting network is somewhat give a look of insertion sort.

In our proposed network a, b, c….......h are input data elements and A, B, C…..F represent the set of comparators which we call part. Part A contains (n-1) comparators; part B contains (n-2) and the last part will contain two comparators. This shows that the Avi network uses n(n-1)/2 comparators where n is the number of elements. In this network each element is compared with next to next element (e.g. part A of Fig 2, 1st element is compared with 3rd element and 2nd is compared with 4th one). This process will go on. On reaching last element (at depth 6 shown in Fig1), we compare last two data element. This A part guarantees that last data element is the largest among all elements. Similarly in part B the same process will be follow as was followed in part A and this works parallel with A and now 2nd last line will be the terminating line for part B as shown in Fig 2. Part B guarantees that 2nd last line has the 2nd largest element. Same procedure will be repeated for part C, D and E which will give largest data element on 3rd, 4th and 5th line respectively as shown in Fig 2. Last comparator of each part; find the largest element of the respective data line. In each network last part will give three sorted elements as do part F in our case. Part F sort first three (1st, 2nd and 3rd) data elements in the ascending order. After executing part F we get the final output of the whole network which gives data in the ascending order.

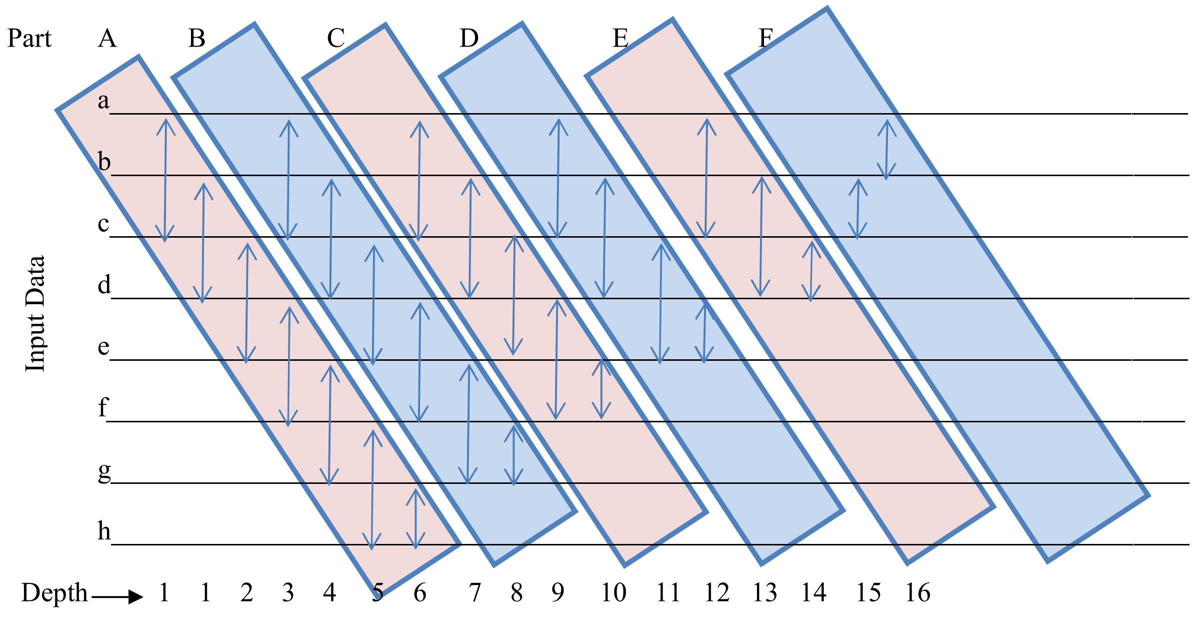

Fig.2 Avi Sorter

III. Explanation in Detail

As we see in Table II, there are total 8 data elements which we want to sort. First column shows those data elements numbers and First row of the table represents the iteration of the network having name I1, I2, I3, I4……………………I15,I16 and last column (Result) gives the sorted output in ascending order. In each column the same colour represents comparator and the swapping of data is shown with red text. The final position of data is shown with green colour. As we see from Fig 2, after execution of part A or which is at depth 7 (I7) largest element of the data gets its final position. In the same way part B (Fig 2) gives the 2nd last largest element on 2nd last line. Similar way process goes on till the whole data in sorted order. Here we see last part F (Fig 2) sort the first three data elements.

IV. Results

There are many sorting network like Odd-Even sorter and Insertion/Bubble sorter [5]. We use both sorting network to compare the performance with Avi network.

TABLE III
Sorting Network Comparators

S. No.	No. of data elements	No. of Comparators		
		Odd-Even Sort= n(n-1)/2	**Insertion Sort = n(n-1)/2**	**Avi Sort = (n(n-1)/2)-1**
1	4	6	6	5
2	6	15	15	14
3	8	28	28	27
4	10	45	45	44
5	12	66	66	65

TABLE II
Working of Avi Sort

S. No.	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13	I14	I15	I16	Result
1	65	45	45	45	45	45	14	14	14	14	14	14	14	14	14	14	10
2	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	14
3	45	65	65	65	65	14	45	45	45	45	45	45	45	45	35	35	35
4	55	55	55	55	55	55	32	32	32	35	35	35	35	35	45	45	45
5	99	99	99	99	14	65	65	65	65	65	55	55	55	55	55	55	55
6	88	88	88	88	88	32	55	55	55	55	65	65	65	65	65	65	65
7	14	14	14	14	99	99	88	88	88	88	88	88	88	88	88	88	88
8	32	32	32	32	32	88	99	99	99	99	99	99	99	99	99	99	99

Table III shows number of comparator of various sorting network. Fig 3 represents the values as given in Table III.

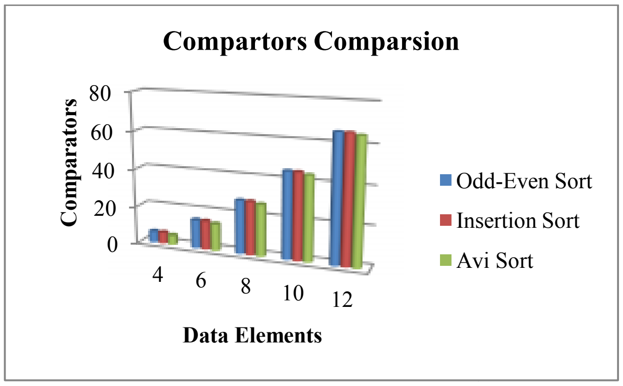

Fig.3 Comparators Comparison

Fig. 3 shows that Avi sorting network has less comparator than Odd-Even and Insertion Sort. Fig. 4 shows depth of Avi sorting network has more depth than other two.

Similarly Table IV shows that depth of various sorting networks as mentioned above.

TABLE IV
SORTING NETWORK DEPTH

S. No.	No. of data elements	Actual Depth of Network		
		Odd-Even Sort= n	Insertion Sort= (2n-3)	Avi Sort= (3n-7)
1	4	4	5	5
2	6	6	9	11
3	8	8	13	17
4	10	10	17	23
5	12	12	21	29

Fig 4 shows the graphical representation of the values as given in Table IV.

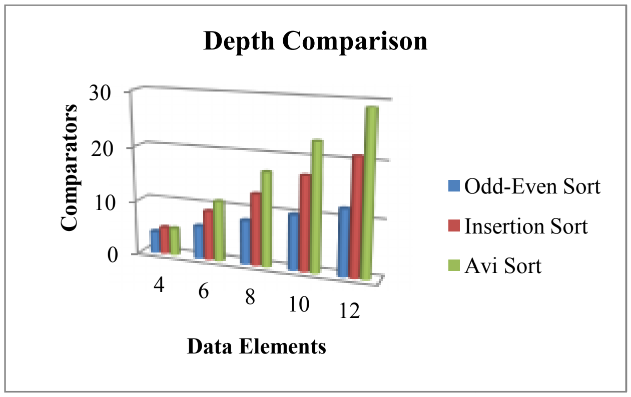

Fig.4 Depth Comparison

Fig. 4 shows that Avi sorting network has more depth than Odd-Even and Insertion Sort. By which we say that this method actually taken larger time as compared to said above.

Table V represent the special case of Avi sorting network. In which if the number of data elements is 2 or 3 then the depth of the Avi network does not follow the formula (3n-7) as shown in Table IV.

TABLE V
AVI SORTING NETWORK FOR DATA ELEMENTS 2& 3.

S. No.	No. of data elements	No. of Comparators	Actual Depth of Network
1	2	1	1
2	3	3	3

V. PROOF

There are N! permutations of numbers in an N-wire network, and to test all of them would take a significant amount of time, especially when N is large. The number of test cases can be reduced significantly, to 2^N, using the zero-one principle. This will be much smaller than N! for even modest N [1]. By applying zero-one principle [3] on the Avi sorting network, we found that it works correctly for all 2^N possible input sequence those are made by the combination of 0 and 1.

VI. CONCLUSION

The main benefits of this network are that it is easy to design and to understand the concept of sorting network and this network reduces comparator as compared to Odd-Even and Insertion sort network. The drawbacks of this network are that, this network has O(n $\log_2$ n) time complexity same as the insertion sorter have and the depth (3n-7) of the network. In future work, we shall try to decrease the depth steps as well as their time complexity by modifying it or designing new sorting network.

REFERENCES

[1] Sorting Network. Available: http://en.wikipedia.org/wiki/Sorting_network.

[2] Andreas Morgenstern and Klaus Schneider, *Synthesis of Parallel Sorting Networks using SAT Solvers,* University of Kaiserslautern, pp 71-80, MBMV 2011

[3] Cormen, T.H., C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms*, MIT Press, 1990.

[4] AKS83 Ajtai, M., J. Komlos, and E. Szemeredi, *An O (n log (n)) sorting network, in Symposium on Theory of Computing (STOC)*, pp 1–9, ACM, 1983.

[5] Insertion Sort. Available: http://en.wikipedia.org/wiki/Insertion_network.

[6] Avinash, Kamal, Shalini, Neha, *Linear Search based Sorting Network,* ICACCT-2012

[7] Avinash Bansal, *Aviva Sorting Network,* IEEE Explore NERIST, ICPEN Dec. 2012