IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 04, Issue 03 (March. 2014), ||V6|| PP 18-23

Freeze Sorting Algorithm Based on Even-Odd Elements

Sarvjeet Singh, Surmeet Kaur
[1] *M.Tech. Scholar,Department of Computer Science and Engineering*
[2]*Asst.Prof.,Department of Computer Science and Engineering*
[1,2]*Lovely Professional University,Phagwara,Punjab,India*

Abstract: - An algorithm is basically term vital part of Operations Research Methodology. There are many algorithms related to sorting, basically it is the operation of logically arrangement of records or elements and it can be used for the numerically data or alphabetically data. Sorting operation implemented in the Data Structure to make efficient searching of elements. The Sorting Algorithm, having the swap, comparison and assignment operations related to the direct complexity of an algorithm. In proposed sorting algorithms we used strategy of selection of the elements i.e. we have the functions called as MinOddFunction(collect the minimum odd value of list) and MinEvenFunction(collect the minimum even valued from list), these functions collect the values from list and compare with each other and freeze it and so on till array is empty. This algorithm has the time complexity in the worst case is O(n^2), where n is the size of data being sorted. In this proposed paper, we will conclude the time complexity on the basis of number of iterations, comparisons, memory time and other factors.

Keywords: - *Best Case, Worst Case,CPU,RAM*

I INTRODUCTION

The proposed paper represents a new method of the sorting of integer data, basically sorting is an logical arrangement of the data according to the requirement of user like ascending or descending of data. The analysis of algorithm having the basically two parts:-
Time Complexity (time taken by hardware to implemented algorithms)
Space complexity (how much space required for running the algorithm).
Complexity in general, measures algorithms efficiency in internal factors such as the time needed to run an algorithm. Sorting is used to make searching efficiently and to improve the performance of CPU. There are two types whereas sorting algorithm requires. First is internal sorting includes collected data which is fitted to the main memory and second is external sorting includes all collected data is not fitted in main memory but is required in the secondary memory. This dissertation report presents a new sorting algorithm and this approach of Divide and Conquer strategy. We use the techniques for analysis of an algorithm called as “Big O Notation”.
Big O analysis of algorithm can be based on:-

1) No. of arithmetic operations in the algorithm.
2) No. of comparison in particular pass.
3) No. of times through a critical loop.
4) No. of array elements accessed.

II BACKGROUND AND EXISTING SORTING ALGORITHMS

1.1 Factors affecting the algorithms:-

1) Computational complexity having the three cases when your particular element search at first time then it is called as best case and when it is at end then is called as worst case and when element takes the time having between of array it is called as average case. In the different cases, complexities will be O(n log n) , O(n^2) and O(n). There are located in middle element case it is always taken the O(n log n) comparison.

2) No. of swaps (for stable and in-place algorithms).

3) Stability: Any algorithm is called as stable sorting algorithm having the element which is having repeated again and again in a single list. Any sort algorithm is called as stable if there is a single list and there has a two same element.for Example: a{1,2,4,5,2,8,3). In this case there are having the same elements and there are having priority to the first element in sorting.

In case of hardware memory and CPU, other thing related to the directly to hardware effected the performance of the algorithm whenever there are we use the in place there are having the O(1) or O(n log n) comparison sort the list if the we having the large data set then we use the cache memory to store the data set as a temporary memory.

4) Recursion: It is the having the important feature whenever the algorithm iterates itself again and again then it is called as the Recursive Process Algorithm. Example Merge Sort.

1.2.1 Classification of algorithms as follows:-

1) Swap Based:- In which pair of element is being exchanged, exp. Shell Sort.
2) Merge Based:-Having sequence swaped with each other afterwards any element, exp. Insertion Sort.
3) Tree Based:-Wherever data is stored in form of binary tree exp, Heap Sort.
4) Other Categories:-Having an additional key for perform swap, Radix and Bucket Sort.

1.2.2 Methods of Sorting is as Follows:-

1) Selection Sorting:- Selection Sort, Heap Sort, Smooth Sort, Strand Sort, Insertion Sort.
2) Insertion Sorting:- Shell Sort, Tree Sort, Liberary Sort.
3) Exchange Sorting:-Bubble Sort, Cocktail Sort, Gnome Sort, Comb Sort, Quick Sort.
4) Merge Sorting:- Merge Sort.
5) Non Comparison Sorts:- Radix Sort, Counting Sort, Bucket Sort
6) Other Sorting techniques:- Topological Sorting, Sorting Network.

1.2 Existing Sorting Algorithms:-

1.2.1 Bubble Sort Algorithm:- This is simplest sorting algorithm, in which elements scan from starting index of an array and the compare with adjacent element if its greater than it swap if it's not than second element compare with third and so on till array is not equal to null, and this process repeated until list is being sorted. The time complexity is O(n) in best case and $O(n^2)$ in worst case and as well as space complexity is depends upon the number of inputs. Bubble Sort Algorithm is as follows:-

Figure 1.1 Bubble Sorting Algorithm

```
BubbleSortAlgorithm( A , n)
1. for i ← 0 to n – 1 do
2.          for j ← 0 to n – i
3.                    if A[j] < A[j+1] then
4.                    swap( A[j] , A[j+1] )
5.          end for of step 2
6. end for of step 1
```

1.2.2 Selection Sorting Algorithm:-In this method of sorting , selection method is being used , basic formula to conclude the algorithm is that smallest element is selected from the array and placed in the proper location of the array in particular pass, this is repeated until list is being sorted. The time complexity of the algorithm is $O(n^2)$ in best and worst case of the input data size.

Figure 1.2 Selection Sorting Algorithm

```
SelectionSortAlgorithm(A,n)
1. i ← 0
2. while i < n do
3. j ← i+1
4.          while j < n do
5.                    if A[i] > A[j] then
6.                              Swap(A[i], A[j])
7.                              j ← j+1
8.                     End if of step 5
8.           End While of step 4
9. i = i+1
10 end-while
```

III PROPOSED SORTING ALGORITHM

In this paper, Freeze Sorting Algorithm is presented proposed algorithm is as follows:-

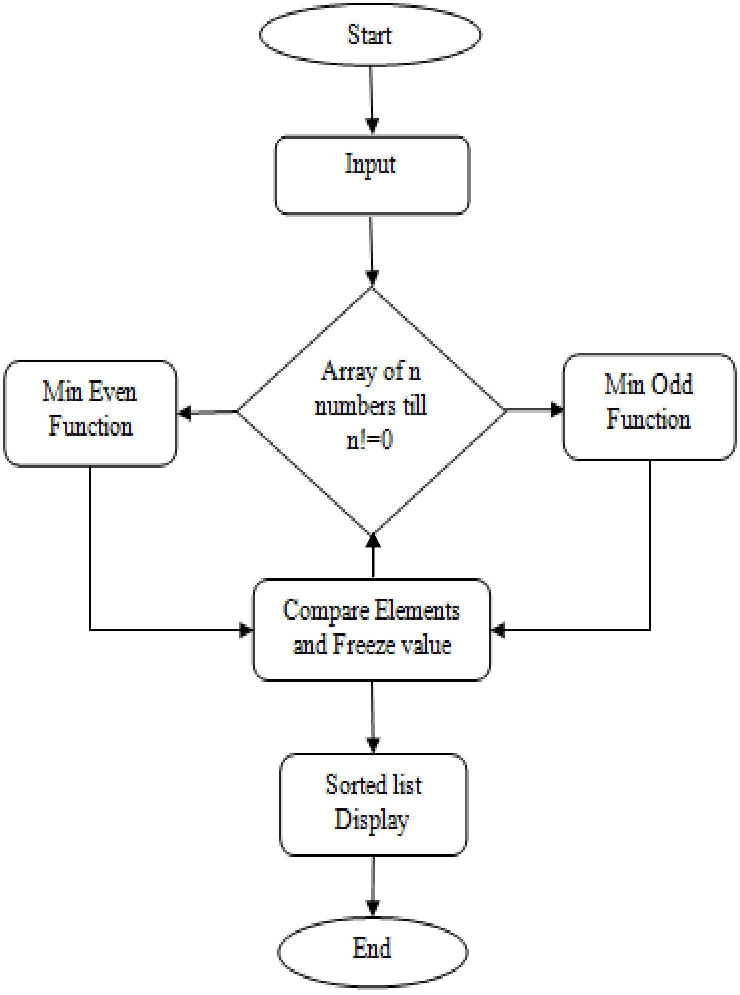
Proposed Algorithm:-

Step 1:- Input the n number of the elements in array.

Step 2:- Indexed the all the elements according to the input.

Step 3:- MinEvenfunction (select the minimum even element from the list) and MinOddFunction(select the minimum even element from the list).
Step 4:- Compare the values selecting from the MinEvenFunction and MinOddfunction and freeze in the list.
Step 5:- Repeat the process until sorting of list.
Step 6:- End

Figure 1.3 Flow Chart of the Algorithm



Computational analysis of proposed algorithm:-
T(1) = (n-2)
T(2) = (n-2)+(n-4)
T(3) = (n-2)+(n-4)+(n-6)
:
:
:
:
:
T(M-1) = (n-2)+(n-4)+(n-6)+………+(n-(n-2))
Where M=n/2
If n is odd number,
$(M) = \lceil (n2\ ) \rceil 2 - ((2\lceil (n2)\ \rceil) - 1) m < n/m=1$
If n is even number,
$(M) = (n2\ )2 - n2\ m < n/2m=1$
So,
$T(M)=O(n^2)$
Example of the propoped algorithm

14	7	12	5	11	10	8	6	3	9	13	4

After First Pass:-

14	7	12	5	11	10	8	6	**3**	9	13	**4**

3	**4**	14	7	12	5	11	10	8	6	9	13

After Second Pass:-

3	4	14	7	12	**5**	11	10	8	**6**	9	13

3	**4**	**5**	**6** ‖	14	7	12	11	10	8	9	13

After Third Pass:-

3	4	5	6	14	**7**	12	11	10	**8**	9	13

3	**4**	**5**	**6**	**7**	**8** ‖	14	12	11	10	9	13

After Fourth Pass:-

3	4	5	6	7	8	14	12	11	**10**	**9**	13

3	**4**	**5**	**6**	**7**	**8**	**9**	**10** ‖	14	12	11	13

After Fifth Pass:-

3	4	5	6	7	8	9	10	14	**12**	**11**	13

3	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12** ‖	14	13

After Sixth Pass:-

3	4	5	6	7	8	9	10	11	12	**14**	**13**

3	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14** ‖

Sorted List As Follows:-

3	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**

IV COMPARISON RESULTS WITH EXISTING ALGORITHMS

1.1 Practical Results:-

In the comparisons of the other algorithms execution time is calculated. We conclude the result from the selection, bubble,insertion,freeze sorting algorithm. The algorithm efficiency is measured on the basis of the CPU time based on the time taken by the running process at background area of the n number of inputs data. The algorithm was obtained results from the C Language with the help of by in built function Clock().It runs on Sony Vaio E-Series laptop with the following specification:-Intel(R) Core™ i5-2450M CPU at 2.50GHz with the 4 GB RAM. In first table result will shows of the clock() function.

Table 1.1 Comparisons of Random Inputs

DATA SIZE(n)	**Bubble Sort(ms)**	**Selection Sort(ms)**	**Freeze Sort(ms)**
10	.686	.602	.596
20	7.36	6.74	6.69
50	23.47	21.34	22.36

100	85.05	78.22	74.53
200	255.45	245.87	226.35

1.2 Comparisons on the basis of complexity and other factors:-

	Bubble Sort Algorithm	Selection Sort Algorithm	Insertion Sort Algorithm	Merge Sort Algorithm	Freeze Sorting Algorithm

Time Complexity:-
Best Case
Average Case
Worst Case
	
	O(n)
	O(n^2)
	O(n^2)
		
		O(n^2)
		O(n^2)
		O(n^2)
			
			O(n)
			O(n^2)
			O(n^2)
				
				O(n log n)
				O(n log n)
				O(n log n)
					
					O(n^2)
					O(n^2)
					O(n^2)

Steps used in above Example	60	7	8	7	6
Space Complexity	O(1)	O(1)	O(1)	O(n)	O(1)
Method used	Exchange	Selection	Incremental	Merging	Selection
In Place Algorithm	Yes	Yes	Yes	No	Yes
Stable Algorithm	Yes	No	Yes	Yes	Yes
Type	Internal Memory	Internal Memory	Internal Memory	Internal and External Memory	Internal Memory

V CONCLUSION

In this proposed algorithms , It has O (n2) complexity. The number of comparisons in particular pass is occur, comparisons is lesser than the Selection sort algorithm. It is called as stable and in place algorithm also because it selects the element from left to right in array. It needs only O (1) space complexity. So the performance of Freeze Sorting algorithm based on the even odd elements is faster than the existing sorting algorithm. This is proved by analytical and experimental point of view.

REFERENCES

[1] Knuth D.E,"*The art of programming- sorting and searching*,".2nd edition Addison Wesley.

[2] Sultanullah Jadoon , Salman Faiz Solehria, Prof. Dr. Salim ur Rehman, Prof. Hamid Jan , "*Design and Analysis of Optimized Selection Sort Algorithm*" International Journal of Electric & Computer Sciences IJECS-IJENS Vol: 11 No: 01, February 2011.

[3] Sultanullah Jadoon, Salman Faiz Solehria, Mubashir Qayum, "*Optimized Selection Sort Algorithm is faster than Insertion Sort Algorithm: a Comparative Study*" International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 11 No: 02, April 2011.

[4] D.E.Knuth, Sorting and Searching, volume 3 of "*The Art of Computer Programming*". Addison Wesley, Reading, MA, (1973).

[5] Wang Min "*Analysis on 2-Element Insertion Sort Algorithm*", International Conference on Computer Design And Appliations (ICCDA), 2010.

[6] Kaur Surmeet,"*Freezing Sort*",International Journal of Applied Information Systems (2249-0868), Volume 2,No.4 May 2012.

[7] Jehad Alnihoud and Rami Mansi," *An Enhancement of Major Sorting Algorithms*," The International Arab Journal of Information Technology, Vol.7.(2010).

[8] Muhammad Anjum Qureshi," *Qureshi Sort: A new Sorting Algorithm*,".(2010).

[9] Cormen T., Leiserson C., Rivest R., and Stein C., "*Introduction to Algorithms*," McGraw Hill, (2001).

[10] Singh Tarundeep, Kaur Surmeet, Kaur Snehdeep," *Enhanced Insertion Sort Algorithm*", international Journal of Computer Applications (0975-8887) ,Volume 64,No.21, February 2013.

[11] Seymour Lpischutz, G A Vijayalakshmi Pai (2006) "*Data Structures*", Tata McGraw-Hill Publishing Company Limited, p.4.11, 9.6, 9.8.

[12] You Yang, Ping Yu, Yan Gan, "*Experimental Study on the Five Sort Algorithms*", International Conference on Mechanic Automation and Control Engineering (MACE), 2011.

[13] Agarwal Aayush, Vikas Pardesi, Namita Agarwal," *A New Approach To Sorting: Min-Max Sorting Algorithm*", May-2013 IJERT.

[14] Bhardwaj Shagun," *Mark-Set{} Sort*",2013 IEEE.

[15] Devi S. Gayathri , K. Selvam , Dr. S. P. Rajagopalan, " *An Abstract to Calculate Big O Factors of Time and Space Complexity of Machine Code*",2011 SEISCON.

[16] Khairullah Md.," *Enhancing Worst Sorting Algorithms*", July-2013, IJAST.

[17] Maurya V. N., Bathla R. K., Kaur Diwinder, Maurya Avadhesh, Gautam Ram Asrey, "*An Alternate Efficient Sorting Algorithm Applicable for Classification of Versatile Data*", April-2013 IJMMAC.

[18] Sareen Pankaj, "*Comparison of Sorting Algorithms (On the Basis of Average Case)*", March 2013 IJRCSSE.