Odd Even Merge Sort based on INDEX

Vuppulanchi Pravalika[1], Lt. K. Ravindra Babu[2], R. DileepKumar[3], M.Thirupathi Reddy[4], Ph.D
[1] Graduate (CSE) Mancherial, Adilabad, AP, [2]Associate Professor in CSE, KITS(S)
[3]Assistant Professor, TKR College, Meerpet, Hyderabad, [4]Principal, AZCET, Mancherial, AP,

ABSTRACT
There are various number of sorting techniques that are used to sort 'n' numbers but time complexity is high in some sorting techniques. The main theme of this paper is to sort a set of given numbers in an effective manner. Using Odd Even Merge Sort the time taken to sort given 'n' numbers will be less. Index will be initially assigned to given numbers, based on the index, the numbers will be shuffled and sorted.

General Terms
Sorting, Sub list, Index

Keywords
Odd Even Merge Sort, bubble sort, selection sort, insertion sort, quick sort, merge sort

1. INTRODUCTION
Sorting is one of the most common applications in computer science the process through which data are arranged according to their values. We are surrounded by data. If the data is not in order, we would spend hour trying to find a single piece of information. Imagine the difficulty of finding some ones telephone number in a telephone book that is not ordered in name sequence, In the same manner if it is the case of dictionary, the complexity of finding the word becomes typical if the dictionary is not according to the alphabetical order. One programming concept common to sorting algorithm is swapping of data between two elements in a list.

Odd Even Merge Sort is used to sort a set of numbers in efficient manner. The Odd Even Merge Sort is distinct from all the existing sorting techniques. This sorting technique is an extent to merge sort. Actually sorting is to arrange the elements in an order. We are having many sorting techniques such as Bubble sort, Selection sort, Insertion sort, heap sort, Quick sort, merge sort etc.

2. EXISTING SYSTEM
Sorting techniques play an important and efficient role in sorting numbers, characters, strings. There are different traditional sorting techniques we are using so far such as Bubble, Insertion , Selection, Merge, Heap Quick etc., Now Let us have a brief discussion about each sorting individually.

In the **Divide and Conquer sorting** paradigm that is used to solve all kinds of computational problems, including Fourier transforms and matrix multiplication. It is essential in the special case of recursion in which the problem is divided into two or more sub problems of exactly the same type, and the solution to the corresponding problem is expressed in terms of the solutions to the sub problems. Divide and conquer strategy helps in simplification of complicated problems, such as Towers of Hanoi and also reduces the runtime of procedure since the sub problems can be simultaneously solved and at the same time it even requires low memory space. In this paradigm, the average number of steps is less when compared to other procedures.

In the **Bubble sort**, the list is divided into two sub lists, sorted and unsorted. The smallest element is bubbled from the unsorted sub list and moved to the sorted sub list. After moving the smallest element to sorted list, the wall moves one element ahead, leads to increase number of sorted elements and decrease number of unsorted ones, if this process is continued once then one sort pass is completed.

In the **Selection sort,** the list is divided into two sub lists, sorted and unsorted by an imaginary wall. We find the smallest element in the list and swap it with the element at beginning of the list. After each selection and swapping, the wall between sub lists move ahead, increasing number of sorted elements and decreasing the number of unsorted elements, if this process is continued once then one sort pass is completed.

In the **Insertion sort,** the list is divided into two sub lists, sorted and unsorted lists. In each pass, the first element of the unsorted sub list is picked up and transferred into sorted sub list by inserting that specific element in appropriate place. This is the most common technique used by card players, they pick up each card, they insert it into the proper sequence. Hence, the resultant list is a sorted list.

Let's consider the case of **quick sort** which is an exchange sort, i.e., items swap positions till the entire array is sorted. The principle underlying the quick sort algorithm is to pick one element in the array which is named as pivot and rearrange the remaining elements around it. The elements less than pivot are moved to left of the pivot and at the same time elements greater than pivot are moved to right side of the pivot. Here the pivot acts as partition element divides the original list into two sub-lists and after partitioning , the pivot is in its ultimately correct position, in this way this recursive procedure is continued till the subsequent sub-lists consist of only one element and the entire list gets sorted. The obtained list comprises of elements in sorted form.

Coming to the **merge sort,** it is a divide and conquer external sorting algorithm. The basic procedure involved in merge sort is to divide the list to be sorted into two smaller sub lists and recursively repeat this procedure till only one element is left in the sub list. Later the various sorted sub-lists are merged back to form the parent list. This merging process goes on recursively till the sorted original list is arrived. There are some of the disadvantages in existing system. Some of them are

- Time taken to solve the problem is high.
- Number of iterations is high.

To overcome these limitations a new technique can be used specifically to reduce the time and at the same time it can used to reduce number of iterations to some extent.

3. BRIEF DESCRIPTION OF PROPOSAL

A newly proposed system is Index Based Merge Sort. In this technique, index will be assigned to numbers that are given to sort. The whole processing will be carried based on the index that is assigned to the numbers. Based on index number, even and odd index number's content will be splitted stored in sub-arrays individually. This splitting process is carried till each sub-array contains single element, then the elements will be compared with the corresponding elements and arranged accordingly.

The process involved in this sorting technique is:

- Initially, the given elements are arranged in an array format.
- Index is assigned to all the elements in the array.
- *B*ased on the index value, even and odd index numbers are constructed in "j" and "k" sub-arrays respectively.
- This process is carried until all the elements are splitted into a single element in a single array**.**
- The elements are compared with the corresponding elements and arranged in the arrays.
- This process is performed till all the given elements are sorted.

3.1 Example Problem

Let us consider the Index Based Merge Sort keenly by going thoroughly through an example problem. Initially, let us consider an example consisting of twelve elements in unsorted order.

12 5 9 3 10 8 6 4 1 7 11 2

Initially, index assigned to all the elements that are in unsorted order as shown in figure 1.

$12_{(0)}\ 5_{(1)}\ 9_{(2)}\ 3_{(3)}\ 10_{(4)}\ 8_{(5)}\ 6_{(6)}\ 4_{(7)}\ 1_{(8)}\ 7_{(9)}\ 11_{(10)}\ 2_{(11)}$

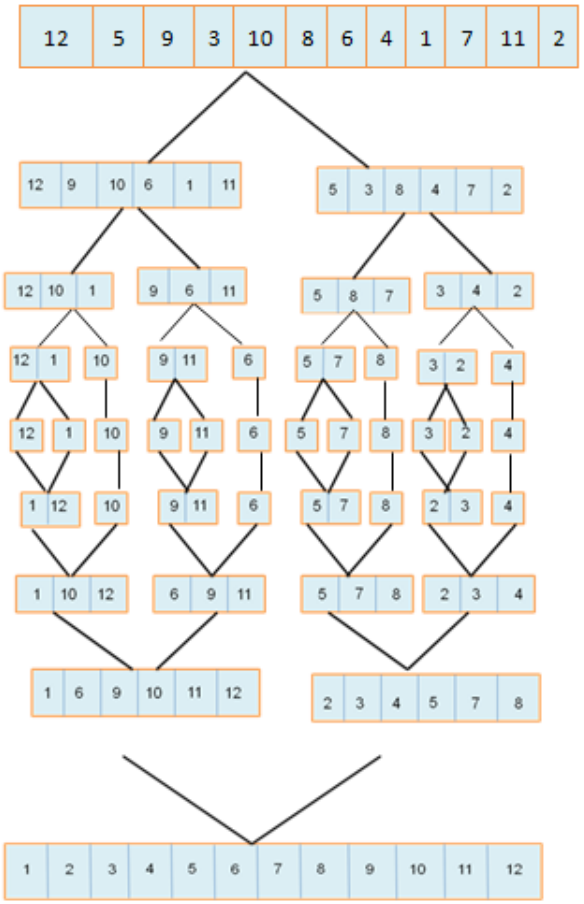

Figure 1: Sequence of steps in proposed method

Let us compare all the different sorting techniques with the new sorting technique i.e., Odd-Even Merge Sort Based on Index.

Advantages of Proposed System

Various advantages using odd-even merge sorting techniques are:

- Time can be saved to a maximum extent.
- Number of iterations can be reduced.
- Risk can be reduced.

4. APPLICATIONS

Odd-Even merge sort can be used in areas as mentioned below and the comparison between various sorting techniques were shown in table 1:

- To arrange numbers in sorted order, if the given numbers are in unsorted order.
- To arrange characters in sorted order.
- To arrange strings in sorted order.

Table 1: Comparison of Various Sorting Techniques.

Sorting Name	Best Case	Average Case	Worst Case	Steps Taken
Bubble-sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	60
Insertion-sort		$O(n^2)$	$O(n^2)$	8
Selection sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	7
Heap-sort		O(nlogn)	O(nlogn)	20
Quick sort		O(nlogn)	O(nlogn)	5
Merge sort		O(nlogn)	O(nlogn)	7
Odd-even merge sort Based on index		O(nlogn)	O(nlogn)	6/7

5. ACKNOWLEDGMENTS

I feel privileged to convey my heartily and sincere thanks to my family for encouraging me. I feel ecstatic to convey my thanks to my friends for supporting me to develop this thought as a paper. Two different sorting techniques can be combined in future to obtain a efficient sorting technique.

6. REFERENCES

[1]C programming and Data Structures padmanabham BS Publications, 2nd Edition, and ISBN: 81-7800-096-2, page no: 42

[2]C programming and Data Structures E.Bala Guruswamy, Mc GrawHill, ISBN (13):978-0-07-0681163 ISBN (10):0-07-0681163, page no: 19.1

[3]C programming and Data Structures Behrouz A.Forouzan Richard F.Gilbey, Technological University series, ISBN-13:978:81-315-1268-5 ISBN-10:81-315-1268-1, page no: 758.

[4]Fundamentals of Computing Algorithms, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Second Edition, and ISBN 978 81 7371 6126, page no: 478.

[5]http://www.codebeach.com/2008/09/sorting-algorithms-in-c.html

[6]http://epaperpress.com/sortsearch/download/sortsearch.pdf

[7]http://www-users.cselabs.umn.edu/classes/Spring-2010/csci5451/FILES/LecN15.pdf

[8]http://faculty.cs.byu.edu/~ringger/Winter2006-CS312/lectures/lecture39-parallelmergesort.pdf

[9]http://www.iti.fh-flensburg.de/lang/algorithmen/sortieren/networks/oemen.htm

[10]C programming with problem solving J.A. Jones and K.Harrow, dreamtech Press.

[11]Programming in C- Stephen G.Kochan, 3rd Edition, Pearson Education.

[12]C for Engineers and Scientists, H.Cheng, Mc.Graw-Hill International Edition.

[13]C and Data Structures - E V Prasad and N B Venkateshwarlu, S.Chand&Co.

[14]C programming &Data Structures, P.Dey, M Ghosh R Thereja, Oxford University Press.